REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	4.Oct.02	DISSERTATION

4. TITLE AND SUBTITLE
EXPLORING LOCATION INFORMATION AND ENABLING ADAPTIVE MOBILE AD HOC NETWORK PROTOCOLS

5. FUNDING NUMBERS

6. AUTHOR(S)
MAJ BOLENG JEFFREY L

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
COLORADO SCHOOL OF MINES

8. PERFORMING ORGANIZATION REPORT NUMBER
CI02-763

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20021017 090

14. SUBJECT TERMS

15. NUMBER OF PAGES
178

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Exploiting Location Information and Enabling Adaptive Mobile Ad Hoc Network Protocols

by

Jeff Boleng

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mathematical and Computer Sciences).

Golden, Colorado
Date Sept. 4, 2002

Signed: _____
Jeff Boleng

Approved: _____
Dr. Tracy Camp
Associate Professor

Golden, Colorado
Date 4 Sept. 2002

pp _____
Dr. Graeme Fairweather
Professor and Head
Department of Mathematical and Computer Sciences

Abstract

Mobile Ad Hoc Networks (MANETs) consist of a group of mobile nodes which form a communication network without prior infrastructure. Each node in the network is responsible to provide services to other nodes in order to realize the ad hoc communication capability. A key component and primary challenge in MANETs is routing data packets over multiple hops between nodes. MANET routing has received considerable research interest recently. However, no routing protocol proposed to date has proven to be effective in the wide range of mobility conditions present in a MANET.

We combine location information and mobility feedback to create an innovative MANET routing protocol which we prove is effective over a wide range of mobility conditions typical in a MANET. We introduce link duration as our mobility feedback metric, and we demonstrate that mobility feedback using link duration effectively enables adaptive MANET protocols. Using our mobility feedback agent, we develop a hybrid MANET routing protocol which adapts in order to combine the strengths of both component protocols while avoiding their weaknesses. Our hybrid, adaptive protocol achieves data packet delivery ratios above 80% in highly demanding network mobility conditions (i.e. link durations less than 4 seconds and node speeds up to 40 m/s). No existing MANET routing protocol can achieve such high performance operating alone. In addition we show that location information increases the performance of MANET routing. Finally, we develop a comprehensive set of mathematical models for data packet delivery ratio, overhead, and delay. These models confirm the suitability of link duration as a mobility metric, validate our simulation results and conclusions, and provide valuable insight into reactive MANET routing protocol operation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview: Ad Hoc Networks

Ad hoc networking involves computers, typically wireless mobile nodes (MNs), that cooperatively form a network without specific user administration or configuration. In other words, ad hoc networking allows an arbitrary collection of MNs to create a network on demand. A node in the ad hoc network, whether it be a laptop, autonomous agent, or sensor, is in charge of routing information between its neighbors, thus maintaining connectivity of the network.

There are numerous scenarios that do not have an available network infrastructure which could benefit from the creation of an ad hoc network:

- rescue/emergency operations: rapid installation of a communication infrastructure during a natural/environmental disaster, or a disaster due to terrorism, that demolished the previous communication infrastructure;
- law enforcement activities: rapid installation of a communication infrastructure during special operations;
- tactical/military missions: rapid installation of a communication infrastructure

in a hostile and/or unknown territory;

- commercial projects: simple installation of a communication infrastructure for commercial gatherings such as conferences, exhibitions, workshops, and meetings;

- educational classrooms: simple installation of a communication infrastructure to create an interactive classroom on demand.

For these reasons, ad hoc networks have been a major focus of research in the last few years (see the reference lists below). We assume the environment for creating an ad hoc network is a broadcast physical medium with limited range, such as the physical medium offered by infrared or radio frequency wireless communications.

There are many challenges in the creation of an ad hoc network: routing challenges (i.e., how to route information to a mobile node that is, perhaps, moving rapidly), wireless medium challenges (e.g., lower bandwidths, higher error rates, more frequent disconnections, and less security than fiber lines), and portability challenges (e.g., lower power and smaller storage capacity than desktop computers). Although experts initially considered only ad hoc networks for a small group of cooperating nodes, many experts now envision very large groups of MNs located over a large geographical area belonging to the same ad hoc network as well. Thus, scalability is another challenge that exists in the creation of an ad hoc network.

Since wireless computing devices are becoming more portable, network-oriented,

and popular, the interest in ad hoc networking is growing. This interest is observable by the recent appearance of numerous proposals for routing in an ad hoc network [12, 14, 15, 22, 33, 42, 43, 46, 48, 53], by the formation of a working group in the Internet Engineering Task Force (IETF) [39][1], and by multiple Internet drafts for routing in a mobile ad hoc network (MANET) [3, 13, 23, 24, 34, 45, 54]. Reviews of unicast routing protocols are provided in [49] and [51], and performance evaluations for some of these protocols are provided in [8], [11], and [32].

Location information has recently been applied to MANET protocols in order to improve the performance of a protocol, to enable scalability, or both [2, 11, 35, 36, 38, 55]. The application of location information has demonstrated performance improvements and promised dramatic scalability [11, 35]. However, the application and inclusion of location information in existing or new protocols is still infrequent. As a result, much of our research involves protocols that use location information.

1.2 Motivation

The IETF MANET working group was formed in 1997 [39]. The working group's charter set December 1999 as a goal to select a standard network routing protocol. As of July 2002, a standard in the form of an accepted Request For Comment (RFC) has not appeared. The lack of a standard cannot be attributed to a lack of research,

[1]The charter of the IETF Mobile Ad Hoc Networks (MANET) Working Group is to develop a solution for routing in an ad hoc network.

focus, or work. Rather it is a result of the inherent difficulty of the problem and the wide range of operating conditions which can occur in a mobile ad hoc network.

In contrast to wired networks, routing in mobile ad hoc networks is challenged by a complicated interaction of three fundamental difficulties. First is contention. The nature of mobile computing devices demands wireless communication. The nature of wireless communication results in significant contention for the shared medium (the wireless channel). Second is congestion. Another aspect of wireless communication is decreased bandwidth which results in much higher congestion when compared to a similar wired network configuration. The links between wireless nodes can support less data traffic than is attainable with wired connections.

Finally, and most importantly, is the unique set of challenges created by mobility. Node mobility in MANETs makes communication links break, and these breaks may occur at a rapid rate. This changing network topology is the key challenge that MANET routing protocols must overcome. Several existing MANET routing protocols have been proposed that deal with this mobility problem in different ways. These protocols and their mechanisms are described in Section 1.3. Any attempt to provide effective routing mechanisms in MANETs must deal with the changing network topology created by mobility.

In addition to mobility, contention, and congestion, MANET protocols must deal with other significant issues. Mobile computing devices are often battery powered and

therefore have limited power and lifetime. They may also be constrained by limited memory or processing capabilities. These additional factors combine with the above three key challenges to make routing in Mobile Ad Hoc Networks extremely difficult.

Our research goals are aimed at improving the effectiveness and scalability of routing in MANETs. More specifically, this research enables MANET routing protocols to adapt their operation based on the current network mobility conditions present. The delivery and use of location information, when combined with adaptive protocols, promises dramatic improvements.

1.3 Related Work

In this section, we categorize the current proposed unicast routing protocols for an ad hoc network. We begin with unicast routing algorithms which do not use location information, then we discuss the inclusion of such information. Details on protocols that directly impact our research are presented in Chapter 2

1.3.1 Unicast Routing Protocols

Many unicast routing protocols have been proposed for ad hoc networks. There are two primary approaches to routing in MANETs, and as a result existing protocols can generally be grouped into one of three categories. The first type of MANET protocols are proactive routing protocols, which include The Wireless Routing Protocol (WRP) [41], Destination-Sequenced Distance Vector Routing (DSDV) [47], Optimized

Link State Routing Protocol (OLSR) [13], Topology Broadcast based on Reverse Path Forwarding (TBRPF) [3], and Fisheye State Routing (FSR) [18]. These protocols proactively maintain network topology through the periodic exchange of control information. In general, proactive protocols are not responsive enough and have too much overhead to be effective when nodes are mobile [8].

The second type of MANET protocols are reactive routing protocols, which include Dynamic Source Routing (DSR) [34], Ad Hoc On-Demand Distance Vector (AODV) [45], and Associativity Based Routing (ABR) [54]. The key to the operation of these protocols is that routes to destinations are only determined and maintained when they are needed (i.e., data is being sent). Unlike proactive protocols, no effort is made to maintain the total network topology. This class of protocols has shown to be more effective when nodes are mobile [51].

The third category of MANET protocols are hybrid in nature. These protocols combine proactive and reactive techniques, and include the Zone Routing Protocol (ZRP) [23] and [24], The Bordercast Resolution Protocol [25], the Temporally-Ordered Routing Algorithm (TORA) [44], and the Landmark Routing Protocol (LANMAR) [17]. As an example of a hybrid MANET protocol, consider ZRP. ZRP defines a zone around each node where the local topology is proactively maintained via the Intrazone Routing Protocol (IARP) [24]. When routes are required outside the local zone, a reactive route discovery mechanism is used via the Interzone Routing Protocol

(IERP) [23].

1.3.2 Unicast Routing Using Location Information

The results presented in [11] show that the use of location information in an ad hoc network significantly improves routing performance of unicast communication; a number of recent MANET unicast routing protocols use location information. Six of these protocols are the Location-Aided Routing (LAR) algorithm [36], the Distance Routing Effect Algorithm for Mobility (DREAM) [2], the Greedy Perimeter Stateless Routing (GPSR) algorithm [35], the Geographical Routing Algorithm (GRA) [31], the Geographic Distance routing (Gedir) protocol [52], and the GRID protocol [38]. A review for some of these protocols is provided in [55].

In these location-based routing protocols, each node maintains a location table that records the location of each other node and the time at which that location information was received. A sender node then uses this information (perhaps combined with a directional antenna [27, 57]) to improve the efficiency in the transmission of packets.

1.3.3 Location Services

Each of the above location-based routing protocols approach the availability of a mobile node's location information differently. For example, knowledge about the location of a destination node is assumed available in GPSR. In fact, in the simulation

results presented in [35], location information is provided to all mobile nodes without cost. DREAM, on the other hand, includes the exchange of location information as a part of the protocol.

The demonstrated success of using location information for routing in an ad hoc network coupled with the recent development of protocols which require location information highlight the need to investigate alternatives to deliver location information. We have developed and evaluated the performance of three location services [10]: the Simple Location Service (SLS), the DREAM Location Service (DLS), and the Reactive Location Service (RLS). These location services are presented in detail in Chapter 4.

An alternative method to provide location information in an ad hoc network is via the Grid Location Service (GLS) [37]. GLS is a hierarchical location service in which each mobile node periodically updates a set of location servers with its current location. The set of location servers chosen is determined by a predefined geographic grid and a predefined ordering of mobile node identifiers in the ad hoc network.

1.4 Research Overview

In this dissertation, we present an innovative MANET protocol which uses both feedback for adaptation and location information to improve routing performance. Our research achieves three primary goals:

1. develop more effective methods for using and delivering location information in MANETs,

2. enable MANET protocols to adapt, and

3. combine the power of protocol adaptivity and location information in a MANET routing protocol.

First we present our comparison of two location aided MANET routing protocols. In addition, we compare their performances with both a popular routing protocol that does not use location information and a basic flooding protocol. We present results and discussion which support the following major conclusions:

1. The added protocol complexity of DREAM [2] does not appear to provide benefits over network wide flooding of data packets.

2. Location information improves DSR [34], especially at high speeds.

3. Promiscuous mode operation improves the performance of DSR significantly.

4. Our implementation of DREAM provides a simple location service.

5. There is a tradeoff between average end-to-end delay and data packet delivery ratio.

The details and support of these conclusions are the subject of Chapter 2.

A mobility metric can be used to enable MANET protocols to adapt. In Chapter 3, we enumerate our requirements for a mobility metric (see Section 3.1), and then discuss several alternative metrics and their adherence to these requirements (see Section 3.2). In general, previously proposed mobility metrics either do not provide good indicators of protocol performance, or require global data from other nodes to be calculated. In other words, previously proposed mobility metrics do not meet our requirements for a mobility metric. One metric, link duration, is shown to satisfy all our mobility metric requirements (see Section 3.3).

Chapters 2 and 3 result in the determination of a set of simulation circumstances that are designed to appropriately test MANET protocols. These simulation parameters have subtle implications which are not initially apparent. Numerous simulation experiments and protocol development has led to a set of MANET simulation "best practices" [4, 6, 7, 10, 11, 56]. Section 3.4 gives the set of simulation parameters which we use in all our research.

The implementation, simulation, and evaluation of different approaches to deliver location information to nodes in a MANET is presented in Chapter 4. Our mechanisms utilize approaches that have parallels with the current approaches to routing in MANETs and wired networks. Specifically, we propose and evaluate (via simulation) three location services for an ad hoc network. One of the three protocols evaluated is a reactive protocol (RLS). The other two protocols evaluated proactively transmit

either location tables to neighbors (SLS) or location information to everyone (DLS). An effective location service can be used to improve the performance and scalability of a routing protocol that requires location information (e.g., GPSR [35]).

We determine that a proactive protocol which periodically floods individual node location information (DLS) is unable to provide accurate location information (especially when mobility is high). On the other hand, a proactive protocol that periodically shares location table entries with neighbors (SLS) offers advantages in terms of simplicity, overhead, and performance.

When our proactive protocol which shares location table entries with neighbors (SLS) is compared with our reactive protocol (RLS), we discover that the flooding requirements of the reactive protocol are much more costly in terms of the number of packets (bytes) that are transmitted over the number of packets (bytes) received. In addition, the percentage of *invalid* location responses provided by the reactive protocol is much higher than the proactive protocol. We, therefore, conclude that our Simple Location Service (SLS) is preferred over both our Reactive Location Service (RLS) and DREAM's Location Service (DLS) (see Chapter 4).

Since we present an effective mobility metric in Chapter 3, the next step is the implementation of a mechanism to measure a node's mobility and provide accurate feedback to the routing protocol. Chapter 5 presents the development of a feedback agent that accurately tracks the mobility of a node in a distributed manner and

provides this information to enable protocol adaptivity. Furthermore, we demonstrate that mobility feedback can be provided with no additional overhead when data traffic is present.

Chapter 6 fuses the use of location information and mobility feedback, which results in a new hybrid, adaptive MANET routing protocol. Our hybrid protocol combines location information and feedback to adapt between two existing MANET routing protocols. We demonstrate the value of using location information, feedback, and adaptive operation via a performance analysis. Our results should encourage others to improve existing protocols or develop new protocols with similar methods.

Our development of a hybrid protocol by combining existing protocols is an effective way to optimize protocol performance over a wide range of network scenarios. We develop an effective hybrid protocol, Adaptive Location Aided Routing from Mines (ALARM), which is superior to both component protocols in data packet delivery and overhead. It combines the low overhead of Location Aided Routing (LAR) [36] in times of mild to moderate mobility with the high delivery ratio of Flood in times of high mobility. ALARM has lower overhead and higher delivery ratios than LAR and Flood for a wide range of mobility scenarios (see Chapter 6). In addition, our feedback agent (see Chapter 5) proves beneficial in dampening the directed flood component of ALARM. Our dampening mechanism allows us to obtain higher deliver ratios with minimal cost.

Finally, mathematical models for reliability, overhead, and delay are presented in Chapter 7. Our modeling results validate link duration as an effective mobility metric. In addition, by comparing simulated performance to predicted theoretical performance, we are able to validate the performance of our hybrid, adaptive routing protocol and gain insight into the nuances of protocol operation and performance. We note that all our simulated results and theoretical results have a correlation coefficient above 0.92.

Chapter 2

EVALUATION OF LOCATION INFORMATION IN UNICAST PERFORMANCE

This chapter provides a detailed, quantitative evaluation comparing the performance of two location based ad hoc network routing protocols: LAR [36] and DREAM [2] (see Section 1.3.2). Simulation results on LAR, DREAM, and other location based protocols exist on the individual protocols; however, since these simulation results are based on different simulation environments, different simulation parameters and even different network simulators, the performances are not comparable. We compare the simulation results for LAR and DREAM with DSR [33], a unicast routing protocol that does not use location information (see Section 1.3.1). We chose DSR since it performs well in many of the performance evaluations of unicast routing protocols (e.g. [8, 32]). We also include simulation results for a basic flooding protocol, which can be viewed as a baseline case.

The NS-2 code used in our simulations of DSR was obtained from [50]; we wrote the NS-2 code used in our simulations of LAR and DREAM. During implementation, we followed the protocol descriptions provided for LAR in [36] and DREAM in [2]. When implementation questions occurred, we contacted the protocol authors for

guidance. We discuss the implementation decisions made and protocol parameters chosen in the description of each protocol. Some of the simulation results we present are different from previously reported results; we discuss the reasons for the differences in this chapter. Lastly, at the end of this chapter, we list five conclusions which summarize our findings.

2.1 Protocols Studied

2.1.1 Dynamic Source Routing (DSR)

Protocol Overview DSR is a source routing protocol which determines routes on demand [33]. In a source routing protocol, each packet carries the full route (a sequenced list of nodes) that the packet should be able to traverse in its header. In an on demand routing protocol (or reactive protocol), a route to a destination is requested only when there is data to send to that destination and a route to that destination is unknown or expired. In the evaluation of DSR, both [8] and [32] only locate routes that consist of bi-directional links. (Although DSR does not require bi-directional links in the protocol, IEEE 802.11 [16] requires bi-directional links in the delivery of all non-broadcast packets.) The version of DSR in our study also only locates bi-directional links. In other words, a route reply packet containing the complete route from source to destination is sent along the reverse route to the source.

An overview of DSR operation is provided by stepping through a simple scenario

in which one node sends a data packet to another. Suppose a source node S needs to send data to a destination D and S does not have a route to destination D available. In DSR, S initiates route discovery by transmitting a route request packet. This packet is first sent to the neighbors of S only (i.e., the time-to-live, or TTL, of the packet is set to zero). If the neighbors of S do not respond with a route for D within a timeout period, then S floods the route request packet in the entire network. The route request packet is retransmitted if S does not receive a response to the packet within another timeout period.

If a node receives a route request packet, does not know a route to D, and the TTL of the route request packet is not zero, the node adds its address to the source route and then forwards the route request packet further. If a node receives a route request packet and does know a route to D, the node sends S a complete route (a sequenced list of nodes) from S to D.

In the version of DSR we use, the Medium Access Control (MAC) layer determines when a link has broken. (Link failures can also be learned from the Network Layer Acknowledgment feature; this feature is not used in our simulation.) Route errors are signaled if an MN times out while waiting to acquire the channel or if an MN does not receive a link layer acknowledgment (ACK) on its transmission. If a packet is dropped as a result of queue or buffer overflows (typically due to congestion), a route error is not signaled. If a route error occurs, the MN that discovers the error

Table 2.1. DSR Constants

Timeout for 1 hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 4 bytes
Buffer size	64 packets
Packet lifetime in buffer	30 seconds
Max rate for route replies	1/second

looks in its cache for another route from itself to D. If another route exists, the MN forwards the packet along the new route. If another route does not exist, the MN drops the packet. Either way, the MN updates its cache of the route error and sends S a route error packet via the reverse source route. To transmit future packets, S either uses another known source route or initiates a new route discovery.

MNs using DSR may operate in promiscuous mode. In promiscuous mode, an MN can learn potentially useful routes by listening to packets not addressed to it. Contrary to comments in [32], we discovered that including promiscuous mode operation in DSR significantly reduced protocol overhead and significantly increased delivery ratio at higher speeds. However, as noted in [32], promiscuous mode operation is power consuming. Thus, we chose to present both promiscuous mode operation and non-promiscuous mode operation in our simulation results for DSR.

Implementation Decisions A version of DSR from [50] was used for our simulations. The constants chosen for DSR's parameters are the same as those used

in [8] and [32] (see Table 2.1). Note that although the time to hold a packet awaiting a route is 30 seconds, the results in [8] and herein never hold a packet for longer than 16 seconds. That is, four packets are transmitted every second and the buffer size for holding packets is 64; thus, no more than 16 seconds of packets can be held.

2.1.2 Location Aided Routing (LAR)

Protocol Overview Like DSR, LAR [36] is an on-demand source routing protocol. The main difference between LAR and DSR is that LAR sends location information in all packets to (hopefully) decrease the overhead of a future route discovery. In DSR [33], if the neighbors of S do not have a route to D, S floods the entire ad hoc network with a route request packet for D. LAR uses location information for MNs to flood a route request packet for D in a *forwarding zone* instead of in the entire ad hoc network. (The term forwarding zone in this section is defined the same as the term request zone in [36].) This forwarding zone is defined by location information on D. The authors of [36] propose two methods used by intermediate nodes between S and D to determine the forwarding zone of a route request packet.

In method 1, which we call LAR Box (see Figure 2.1), a neighbor of S determines if it is within the forwarding zone by using the location of S and the expected zone for D. The expected zone is a circular area determined by the most recent location information on D, (X_D, Y_D), the time of this location information, (t_0), the average

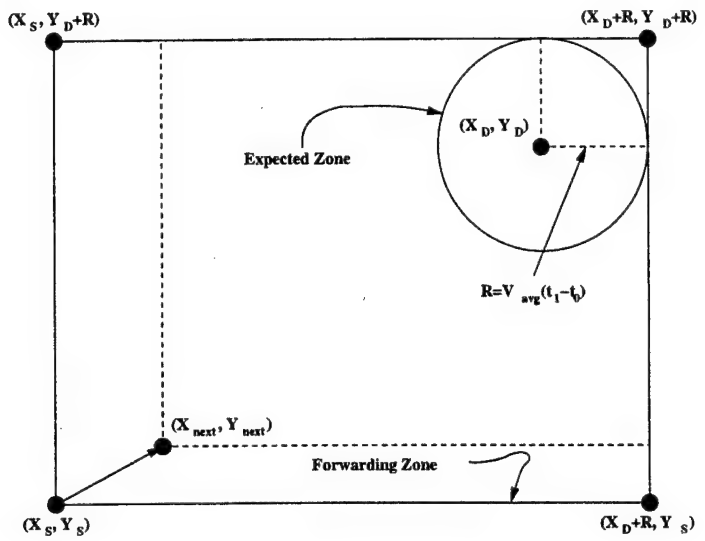

FIG. 2.1. LAR Box Forwarding Zone Technique.

velocity of D, (V_{avg}), and the current time, (t_1). This information creates a circle with radius $R = V_{avg} \times (t_1 - t_0)$ centered at (X_D, Y_D). The forwarding zone is a rectangle with S in one corner, (X_S, Y_S), and the circle containing D in the other corner.

If a neighbor of S determines it is within the forwarding zone, it forwards the route request packet further. An MN that is not a neighbor of S determines if it is within the forwarding zone by using the location of the neighbor that sent the MN the route request packet and the expected zone for D based on the most recent available information. Thus the forwarding zone and the expected zone adapt during transmission; as an example, see (X_{next}, Y_{next}) in Figure 2.1. (This adaptation is mentioned in [36] as a possible optimization to the LAR protocol.)

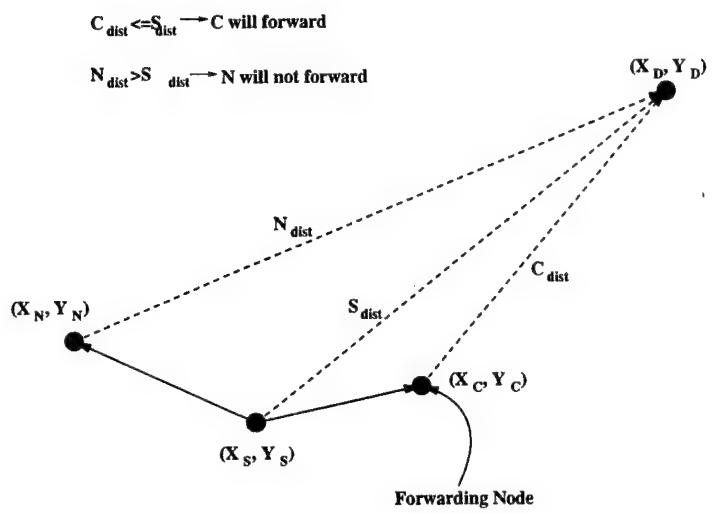

FIG. 2.2. LAR Step Forwarding Zone Technique.

In method 2, which we call LAR Step (see Figure 2.2), an intermediate MN determines if it is within the forwarding zone if the MN is closer to D than the neighbor that sent the MN the route request packet. Specifically, if the distance of the neighbor that sent the MN the route request packet to D is S_{dist}, and the distance of the MN that received the route request packet to D is C_{dist}, then the MN forwards the route request packet if $C_{dist} \leq S_{dist}$.

In both LAR Box and LAR Step, [36] offers the option to increase or decrease the size of the forwarding zone via an error factor, δ. With this error factor, the above formulas become:

$$\text{LAR Box:} \qquad R = (V_{avg} \times (t_1 - t_0)) + \delta$$

LAR Step: $C_{dist} \leq (S_{dist} + \delta)$

Both LAR Box and LAR Step include a two stage route discovery method. In the first stage, the route request packet is forwarded according to either LAR Box or LAR Step. If a route reply packet is not received within the route request timeout period, then a second route request packet is flooded through the entire ad hoc network. If a route reply packet is (again) not received within the route request timeout period, then D is considered unreachable. If D remains unreachable for 30 seconds, packets for D are dropped.

Implementation Decisions Unlike the performance results on LAR presented in [36], we evaluated all the variations and optimizations (except the alternative definitions of the forwarding zone) proposed in [11]. These optimizations include adaptation of the request zone based on more recent location information (discussed above), propagation of location and speed information in every packet transmitted, and local search for route repair. The results presented in Section 2.3 include two of these three optimizations. We did not include the local search optimization (see [36]) in our simulations since the performance results in doing so were unsatisfactory. When an intermediate MN attempts a local search, data packets are held at the intermediate MN (instead of dropped) in the hope that a route discovery by the intermediate MN will prove beneficial. Waiting at the intermediate MN increases end-to-end delay

Table 2.2. LAR Constants

Timeout for 1 hop route request	30 ms
Route request timeout	500 ms
Forwarding error factor (δ)	0.0
Size of header with n addresses	4n + 40 bytes
Buffer size	64 packets
Packet lifetime in buffer	30 seconds

substantially; specifically, if a route isn't discovered, the data packet may wait a full 30 seconds at each intermediate MN that experiences a source route link failure.

In our LAR implementation, as in DSR, a source asks its neighbors for a route to a destination before transmitting a route request in the forwarding zone. Although this feature is not mentioned in [36], we found that including this feature improved the performance results. Lastly, although not mentioned as a possible variation in [36], we evaluated allowing an intermediate MN to respond to a route request (if a route is available). However, a route reply from an intermediate MN does not update the source with recent location information on the destination; thus, the source floods route requests more often when this variation is used. Without allowing an intermediate MN to respond to a route request, the benefits of promiscuous mode operation are significantly reduced. Thus, our performance results on LAR are for non-promiscuous mode operation.

In the LAR protocol, route errors are generated when a route breaks; since a MAC layer does not exist in the original LAR implementation (see [36]), details on

how route errors are generated are missing. In our implementation of LAR, following the implementation of DSR, route errors in LAR are discovered by the MAC layer via link layer feedback at the transmitting node. When a route error is discovered, a route error packet is unicast to S along the reverse source route. Lastly, when a route error occurs, the MN that discovers the error looks in its cache for another route from itself to D. In other words, similar to DSR, the MN forwards the packet along a new route if another route is available. Table 2.2 lists the constants used in our implementation of LAR.

2.1.3 DREAM

Protocol Overview Unlike DSR and LAR, DREAM is not an on demand routing protocol [2]. Instead, each MN in this proactive protocol maintains a location table for all other nodes in the ad hoc network. To maintain the table, each MN transmits location packets to nearby MNs in the ad hoc network at a given frequency and to faraway MNs in the ad hoc network at another lower frequency. Each location packet (LP), which updates location tables, contains the coordinates of the source node based on some reference system, the source node's speed and the time the LP was transmitted. Suppose a source node S needs to send data to a destination D. In DREAM, S first calculates a circle around the most recent location information for D, using the last known speed. The radius is $R = V_{max} \times (t_1 - t_0)$ centered at (X_d, Y_d).

Once the circle is calculated, S defines its forwarding zone (a cone) to be the region enclosed by an angle whose vertex is at S and whose sides are tangent to the circle calculated for D. Similar to LAR, S sends a packet for D to all its neighbors in the forwarding zone; however, in DREAM, the packet is a data packet not a route request. Each of these neighbors then compute their own forwarding zones, based on their own location tables, and forward the packet accordingly. When D receives a data packet, D returns an ACK packet. The ACK packet is sent to S in the same manner as the data packet was sent to D.

An ACK packet may not be received by S due to the following reasons: there is no route to the destination from the source (i.e., no neighbors in the calculated cone), there is no route to the source from the destination, or there is an error in transmission (e.g., a queue overflow due to congestion). If S does not receive an ACK packet within a timeout period, then S resorts to a recovery procedure. In our implementation of DREAM in NS-2, following the work done in [2], the recovery procedure floods the data packet to D. If D receives a flooded data packet, D does not return an ACK packet. Lastly, DREAM defines a timeout value on location information. If the location information is older than the limit specified, then S immediately resorts to the recovery procedure (i.e., flooding).

Implementation Decisions In our first implementation of DREAM, the cone angle was often so small that no neighbors existed in the forwarding zone. Although

Table 2.3. DREAM Constants

Minimum cone angle	30 degrees
Nearby MN defined as within	1 hop
α for nearby LPs	10
X for faraway LPs	13
Y for faraway LPs	23 seconds
LP update offset	0.01 seconds
Location table entry timeout	46 seconds
Timeout for receiving ACK	500 ms

it is not discussed in [2], the simulation results presented there are based on DREAM using a minimum cone angle of 30 degrees [1]. Thus, we added a minimum cone angle of 30 degrees to our implementation of DREAM in NS-2.

We evaluated all the optimizations proposed in [2] for DREAM and also evaluated other variations of the protocol in an attempt to improve the performance of the protocol. In one optimization, an MN transmits location packets (LPs) adaptively based on when the MN has moved a specified distance from its last update location. Although this optimization is proposed in [2], it is not evaluated and details on how to implement this optimization are not provided. Our solution for the transmission of LPs follows:

transmit nearby LP: $\frac{T_{range}}{\alpha} * \frac{1}{\nu} = \frac{T_{range}}{(\alpha\nu)}$,

transmit faraway LP: one for every X nearby LPs;

one at least every Y seconds,

where T_{range} is the transmission range of the MN, ν is the average velocity of the MN, and α is a scaling factor. In our simulations, we set α to 10, X to 13, and Y to 23 seconds. (We optimized these three values via numerous simulation trials.) To avoid LPs being transmitted by MNs at the same time (and, thus, colliding), MNs offset the transmission of their LPs randomly.

In the performance results on DREAM presented in [2], LPs are transmitted periodically and these packets are sent to nearby MNs (100 meters or closer) at a higher frequency than to faraway MNs. (Note that LPs to faraway MNs also update nearby MNs.) We compared sending LPs periodically (as done in [2]) with our solution for the transmission of LPs. We found that our solution reduces the total packets transferred in the simulation by 19% and that the data packet delivery ratio of the two solutions is approximately equivalent (i.e., within 1% of each other).

In the DREAM protocol, nearby MNs are categorized by distance. A variation of the protocol is to specify nearby MNs as being within a given number of hops. We compared the performance of defining a nearby MN as being within 100 meters versus being within one hop and discovered one hop slightly improves the results of the protocol. Thus, unlike the results presented in [2], our implementation of DREAM defines nearby MNs as one hop neighbors.

In the DREAM protocol, each ACK packet is sent to the source via the DREAM protocol. We attempted to reduce the flooding of ACK packets in the forwarding

zone by sending each ACK via the reverse source route, which is gathered by the data packet. While this variation of the DREAM protocol does reduce the total packets transmitted by 6% without a large decrease in the data packet delivery ratio (i.e., the decrease is only 0.8%), this variation of the DREAM protocol adds 11% to end-to-end delay. The increase in end-to-end delay occurs because a unicast ACK is less likely to be delivered than a flooded (in the forwarding zone) ACK; thus, an ACK timeout is more likely to occur. Due to the large increase in end-to-end delay, we chose to not include this variation in our simulation results. Table 2.3 lists the constants used in our implementation of DREAM.

2.2 Simulation Environment

Table 2.4 lists the simulation parameters that we used along with those of [8] and [32] (the random scenarios)[1]. We compare our choices with the choices made in [8] and [32] in order to validate our choices and to illustrate the differences in these three performance investigations of ad hoc network routing protocols. Our main goal was to *stress the protocols* with high data load during both low and high speeds. Our simulation parameters accomplished this goal.

As discussed in [32], a square simulation area allows MNs to move more freely than a rectangular simulation area; however, a square simulation area results in a

[1]See Section 3.4 for more details on our simulation parameters.

Table 2.4. Simulation Parameters

	in [9]	*in* [45]	*herein*
Simulator	NS-2	NS-2	NS-2
Simulation time	900 seconds	250 seconds	1000 seconds
Simulation area	1500x300m	1000x1000m	300x600m
Number of MNs	50	50	50
Transmission range	250m	250m	100m
Average neighbors	11.72	6.32	7.76
Movement model	random way-point	random way-point	random way-point
Maximum speed	1 and 20 m/s	0-20 m/s	0-22 m/s
Average speed	1 and 10 m/s	not specified	0-20 m/s
Pause time	0, 30, 60, 120 300, 600, 900 seconds	1 second	10 seconds $\pm$ 10%
CBR sources	10, 20, or 30	15	20
Data payload	64 bytes	64 bytes	64 bytes
Packet rate	4 packets/second	5 packets/second	4 packets/second
Traffic pattern	peer-to-peer	random	peer-to-peer

smaller average number of hops between the senders and receivers than a rectangular simulation area with the same area (assuming the MNs have the same transmission range). We, therefore, chose to use a rectangular simulation area.

Table 2.4 shows that our simulation area and transmission range are smaller than those used in [8] and [32]. However, if MNs are placed uniformly in the simulation area, and if edge effects are considered (i.e., fewer neighbors exist for those MNs near an edge), then an MN in [8] has an average of 11.7 neighbors and an MN in the random scenarios of [32] has an average of 6.3 neighbors. Our simulation parameters give us an average of 7.7 neighbors. In other words, although our simulation area and

transmission range are smaller than those used in [8] and [32], the environments in these three performance evaluations are similar.

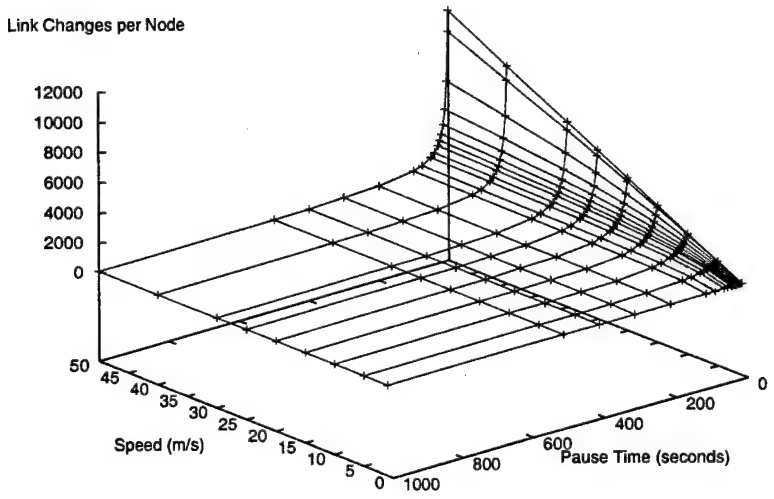

FIG. 2.3. Link Breakage vs. Speed vs. Pause Time.
100m transmission range, 300x600m area, 50 mobile nodes.

In each simulation, we have 50 MNs moving according to the random way-point model [8]. In this model, each MN randomly selects a destination and then moves toward that destination at a given speed. Once the destination is reached, the MN pauses for a given pause time. The MN then selects another destination and repeats the above behavior. With this mobility model, there is a complex relationship between node speed and pause time. For example, a scenario with fast MNs and long pause

times actually produces a more stable network than a scenario with slower MNs and shorter pause times. Figure 2.3 illustrates that long pause times (i.e., over 50 seconds) produce a stable network (i.e., few link changes per MN) even at high speeds [4]. In other words, even though our simulations run for 1000 seconds, the figure indicates that the network is pretty stable for all pause times over 50 seconds. Thus, we chose to keep the pause times short and to vary speed along the x-axis in all of our simulations.

In our simulations, the speed of an MN between the MN's current location and its next destination is chosen from a uniform distribution between $avg \pm 10\%$ meters per second (m/s), where avg is set to 0, 1, 5, 10, 15, and 20. For example, when our speed is set to 20 m/s, all nodes have speeds between 18 and 22 m/s. In [8], when the speed is set to 20 m/s, the average speed is only 10 m/s. Our narrow range of speeds prevents the creation of a stable "backbone" consisting of a few slowly moving MNs.

Figure 2.4 illustrates the average MN neighbor percentage for MNs using the random way-point model (speed is 1 m/s and pause time is zero) as time progresses. The average MN neighbor percentage is the percentage of total MNs that are a given MN's neighbor. As Figure 2.4 illustrates, there is high variability during the first 600 seconds of simulation time as MNs moving with the random way-point model initially move to (or through) the center of the simulation area. We remove this variability in our simulation results by having the MNs move for 1000 seconds of simulation time before sending any data packets. Thus, when data begins transmitting in the two

FIG. 2.4. Average Neighbor Percentage vs. Time.
100m transmission range, 300x600m area, 50 mobile nodes.
Node speed = 1 m/s, pause time = 0 seconds.

reactive protocols at simulation time 1000, there is no routing state in any of the MNs. As a result, initial route request packets are flooded in the entire network for both DSR and LAR. Since DREAM is a proactive protocol, MNs using DREAM begin sending control packets at simulation time 950 seconds; thus, location information used in DREAM is propagated in the network before data packets begin transmitting. Data packets begin transmitting at 1000 seconds simulation time. Our simulations then execute for another 1000 seconds (until the simulation clock is at 2000 seconds).

Our communication model is similar to the communication model used in [8]

and [32]. Specifically, we have 20 CBR (constant bit rate) sources sending 64 byte packets at a rate of 4 packets per second to 20 receivers. One difference between the communication models is that [32] randomly spreads the traffic among all MNs, while [8] and our simulations create peer-to-peer traffic patterns. Peer-to-peer traffic stresses the network protocols since traffic is concentrated in specific areas of the network. We avoid unnecessary contention in the transmission of packets; we offset the transmission of a data packet by 0.0001 seconds for each of the 20 peer-to-peer communication pairs.

We performed 10 simulation trials for each of six speeds. The same 60 mobility scenarios are used to compare the different routing protocols. At zero speed, we use network configurations that occur after the MNs have moved for 1000 seconds. In other words, we first allow the static MNs to distribute in a fashion that is typical of the random way-point model. In addition, at zero speed, we use network configurations that are not partitioned between the sources and destinations since all protocols fail when the network is partitioned.

2.3 Simulation Results

In our comparison of DSR-P (promiscuous mode), DSR-NP (non-promiscuous mode), LAR-NP Box, LAR-NP Step, and DREAM (which is, by definition, NP), we consider the following performance metrics: protocol overhead, network-wide data

load, end-to-end delay, and data packet delivery ratio[2]. The data packet delivery ratio is the ratio of the number of data packets delivered to the destination nodes divided by the number of data packets transmitted by the source nodes. We compare the performance results of the five protocols with flooding every data packet in the ad hoc network (Flood), which allows us to determine quantitatively how well the five routing protocols do against a baseline case.

In our simulations, Flood and DREAM protocols have the highest average hop count: approximately 4.0 across all speeds. (The average hop count for Flood and DREAM is calculated from the first data packet to arrive at the destination.) DREAM resorts to its flooding recovery procedure often (see discussion of Figure 2.6); thus, the average hop count of DREAM and Flood are similar. LAR Box and LAR Step find routes in a similar fashion; thus, the average hop counts of these two protocols are nearly the same: approximately 3.5 across all speeds. Since the LAR protocols deliver a higher percentage of data packets than DSR (see Figure 2.5), the extra packets delivered by LAR are traveling along longer routes. In other words, the two DSR protocols have the lowest average hop count. DSR-P and DSR-NP have an average hop count of 3.0 at low speeds; at higher speeds, the average hop count for DSR-P drops under 2.8 and the average hop count for DSR-NP drops under 2.3. In other words, both DSR protocols have difficulty maintaining long routes at high

[2]In our discussions below, DSR refers to both DSR-P and DSR-NP and LAR refers to both LAR Box and LAR Step.

speeds.

All the performance results presented are an average of 10 different simulation trials. We calculate a 95% confidence interval for the unknown mean, and we plot these confidence intervals on the figures. Since most of the confidence intervals are quite small (in fact, some of the intervals are smaller than the symbol used to represent the mean on our plots), we are convinced that our simulation results precisely represent the unknown mean.

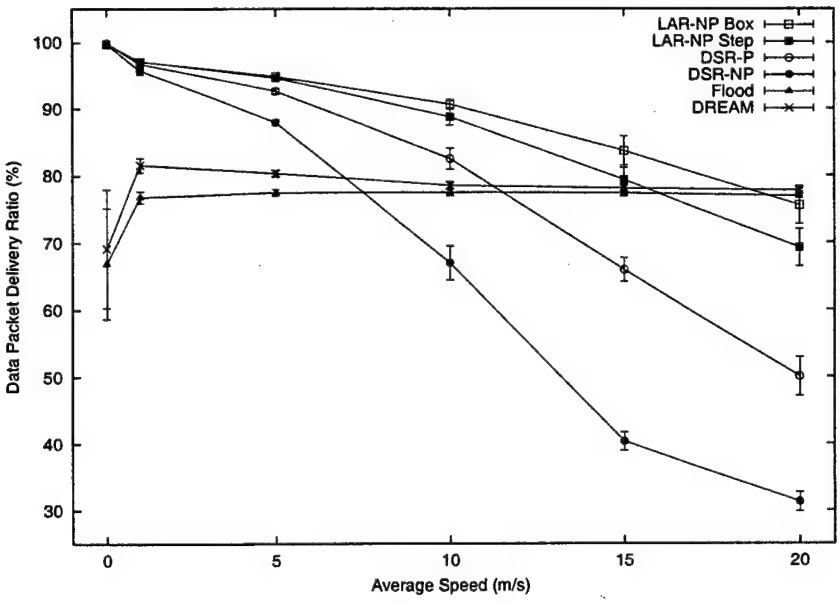

FIG. 2.5. Data Packet Delivery Ratio vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

Performance Figure 2.5 illustrates the data packet delivery ratio versus speed. When speed is zero, the data packet delivery ratios for the DSR and LAR protocols are 100% and the data packet delivery ratios of the DREAM and Flood protocols are approximately 68%. 100% delivery ratio is not achieved by the DREAM and Flood protocols due to the limited buffer size and the contention and congestion in the network caused by the flooding nature of these two protocols. (See Figure 2.6 for a discussion on how often DREAM floods the entire ad hoc network.)

Contention and congestion also contribute to the constant data packet delivery ratio for DREAM and Flood as speed increases from 1 m/s to 20 m/s. In other words, contention and congestion, due to the flooding behavior of these two protocols, override the effect of speed.

In Figure 2.5 for low (or no) speed, the data packet delivery ratios of the DSR and LAR protocols are almost equivalent. As speed increases, however, the data packet delivery ratios of the two LAR protocols are higher than the data packet delivery ratios of the two DSR protocols. When a route is broken from a source to a destination in LAR, the source is able to use location information on the destination to find a new route to the destination more efficiently than DSR's route discovery method.

The data packet delivery rate decreases, as speed increases, for the DSR and LAR protocols. As speed increases, it is much more difficult to find a usable route to

a destination. Figure 2.5 does illustrates that the use of promiscuous mode in DSR significantly aids MNs in learning useful routes.

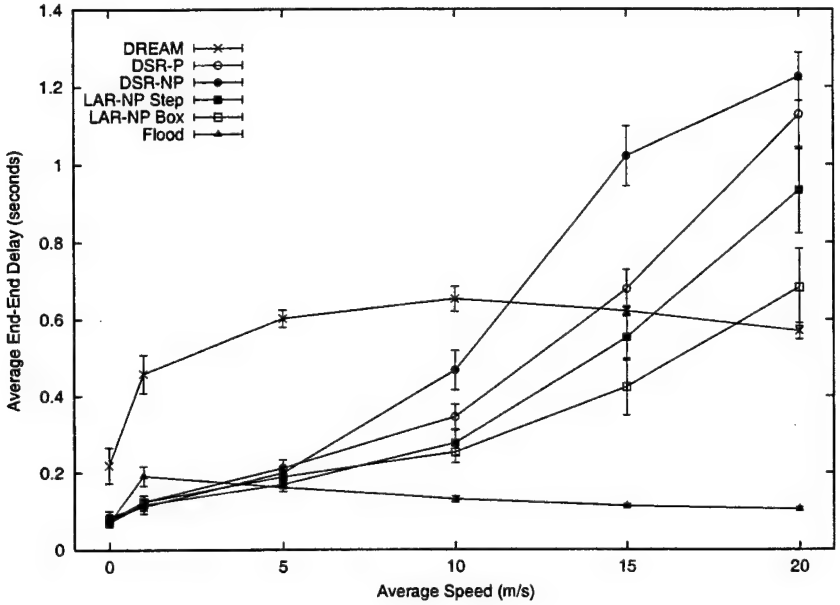

FIG. 2.6. End-To-End Delay vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

Figure 2.6 illustrates the average end-to-end delay of a data packet as speed increases. (The average end-to-end delay of Flood and DREAM is calculated from the first data packet to arrive at the destination.) Since no partitions are included in the zero speed results, the DSR and LAR protocols only need to do route discovery once at zero speed. DREAM also has a good chance of sending data packets without

the recovery procedure at zero speed. Thus, compared to higher speeds, all five routing protocols have a smaller end-to-end delay.

As shown in Figure 2.6, DREAM has the highest average end-to-end delay of all six protocols at speeds less than or equal to 10 m/s. At zero speed, location information on the MNs in DREAM is accurate; however, due to contention and congestion in the network, there is a good chance that a data packet (or an ACK packet for a data packet) does not reach its intended destination. Specifically, the DREAM recovery procedure (i.e., flooding) is used approximately 40% of the time at zero speed. Since a source has a timeout for receiving an ACK of 500 ms in the DREAM protocol (see Table 2.3), the end-to-end delay for DREAM at zero speed is approximately 0.2 seconds. At 1 m/s, the DREAM recovery procedure is used approximately 90% of the time and at 5 m/s and higher, the DREAM recovery procedure is used for almost every data packet transmitted.

As speed increases, more route requests are needed in DSR and LAR; thus, end-to-end delay increases in both protocols as speed increases. The end-to-end delays of DSR are slightly higher than the end-to-end delays of LAR since LAR is (sometimes) able to use location information to focus its search for a route to a destination. At some speeds, DSR-NP has a higher end-to-end delay than DSR-P. A route request in DSR-NP takes longer than a route request in DSR-P, since an intermediate MN in DSR-P may respond to the route request instead of the destination MN.

As shown in Figure 2.6, Flood has (almost) the lowest average end-to-end delay of all six protocols. At low speeds, the average end-to-end delay of Flood is equal to or higher than the average end-to-end delay of DSR and LAR since DSR and LAR spend little time on route discovery at low speeds. At high speeds, however, both DSR and LAR spend time on route discovery; thus, the average end-to-end delay for Flood is lower than the average end-to-end delay of DSR and LAR at high speeds.

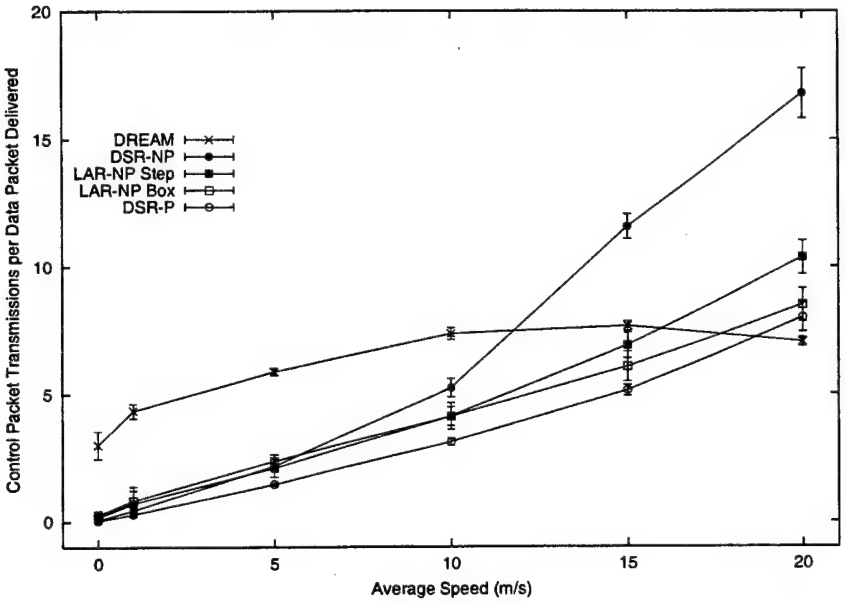

FIG. 2.7. Control Packet Overhead vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

Overhead/Load Figure 2.7 shows the number of control packet transmissions for each data packet delivered as speed increases, which helps capture the power overhead requirements of each protocol. DREAM transmits many small control packets in its exchange of location information. Since DREAM is the only protocol with a proactive element, and the only protocol that returns an ACK for each data packet that is delivered from the forwarding zone, DREAM has the highest control packet overhead at low speeds.

LAR control packet overheads are either equal to or higher than DSR-P control packet overheads. In DSR-P, an intermediate MN responds to a route request if a route is available. In LAR, based on the discussion in Section 2.1.2, the route request is forwarded all the way to the destination before a response occurs. Thus, LAR has the potential of transmitting more control packets than DSR-P.

DSR-NP has higher packet overhead than DSR-P at speeds greater than 5 m/s. An MN using promiscuous mode learns new routes (which sometimes prove to be useful at a later time) from packets not addressed to it. Promiscuous mode operation is more beneficial at higher speeds; thus, the difference in packet overhead between DSR-NP and DSR-P is more pronounced at higher speeds. Higher overhead can be acceptable if the performance (e.g., the data packet delivery rate) is also higher. Figure 2.5 illustrates that this is not the case for DSR-NP.

The control packet overheads of the DSR and LAR protocols increase substan-

tially as speed increases, since more route error and route request packets are transmitted at higher speeds. In DREAM, an ACK is returned by D for each copy of each data packet it receives from the forwarding zone (i.e., not from the recovery procedure). As discussed in Figure 2.6, the recovery procedure is not used often at low speeds; at high speeds, however, the recovery procedure is used often. Thus, the control packet overhead of DREAM increases at low speeds and decreases at high speeds (i.e., fewer ACKs are transmitted at higher speeds).

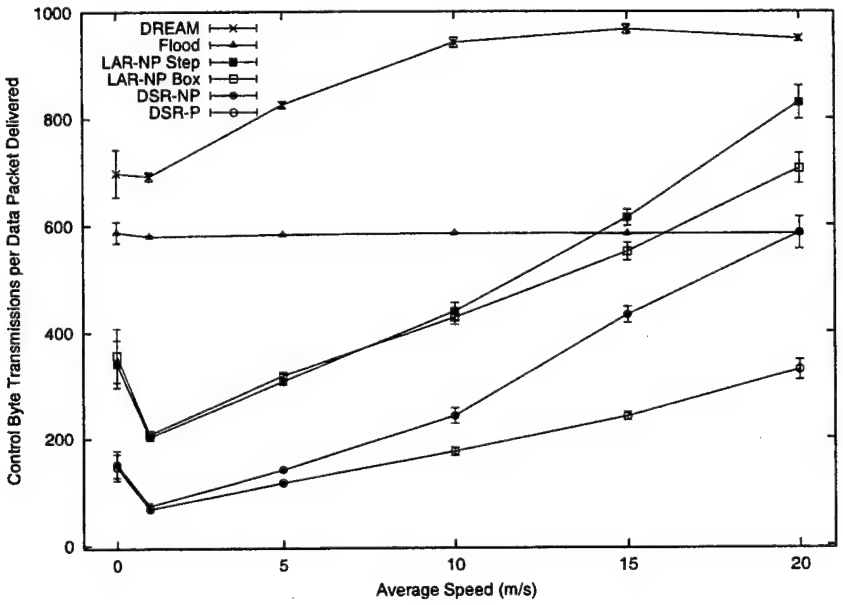

FIG. 2.8. Control Byte Overhead vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

Figure 2.8 illustrates the number of control byte transmissions (in both control packets and data packets) for each data packet delivered as speed increases, which helps capture the bandwidth overhead requirements of each protocol. Both DREAM and Flood have high control byte overhead due to the large number of data packets both these protocols send (see Figure 2.9). Flood has lower control byte overhead than DREAM since Flood does not transmit any control packets; DREAM, on the other hand, transmits many (small) control packets containing location information (see Figure 2.7).

The control byte overheads for the two LAR protocols are higher than the control byte overheads for the two DSR protocols. In addition to transmitting as many (or more) control packets, LAR packets (both control and data) are each 36 bytes larger than DSR packets due to the location information included in each packet. Furthermore, since the average number of hops is larger for the two LAR protocols, the source route included in each LAR packet is larger than the source route included in each DSR packet.

At all non-zero speeds, DSR-NP has higher control byte overhead than DSR-P due to the higher number of control packet transmissions (see Figure 2.7). As speed increases, the control byte overheads of both LAR and DSR increase substantially; in both cases, more route error and route request packets are transmitted as speed increases.

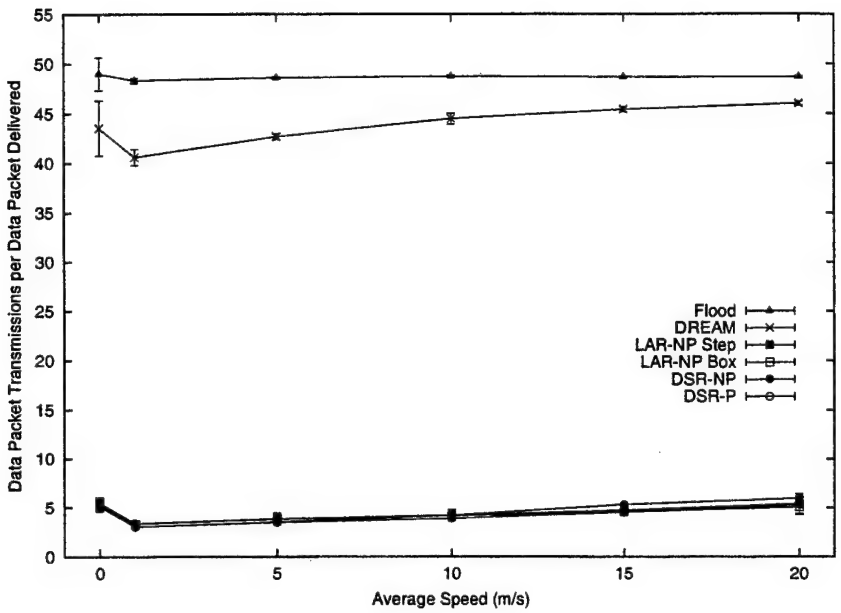

FIG. 2.9. Data Packet Load vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

Figures 2.9 and 2.10 illustrate the data load of the six protocols studies as a function of node speed. In the DREAM protocol, data packets are first flooded in the forwarding zone and then possibly flooded in the entire network. In other words, the DREAM protocol never unicasts a data packet. In our investigation, the DREAM recovery procedure is called between 40-100% of the time (see discussion of Figure 2.6). Thus, both Flood and DREAM have extremely high data load, which is shown in Figure 2.9 for number of data packet transmissions per data packet delivered

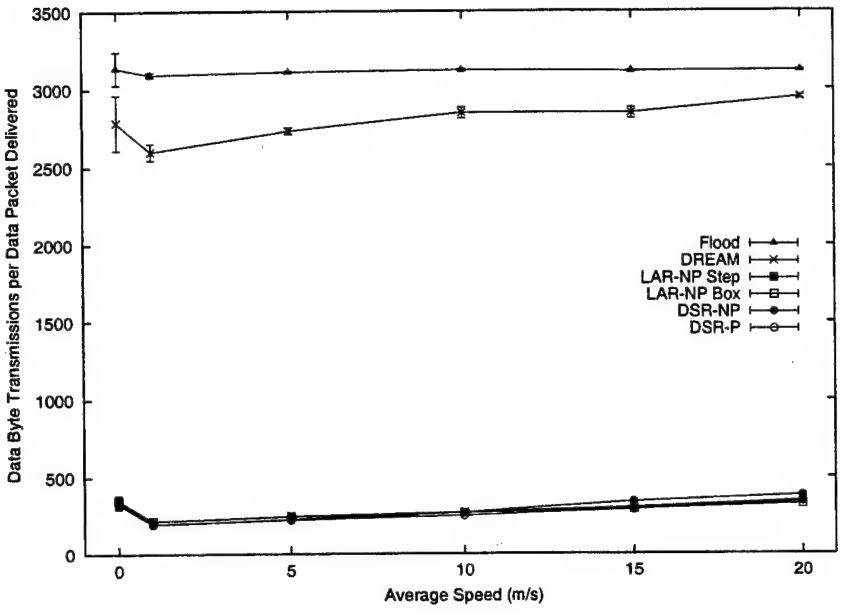

FIG. 2.10. Data Byte Load vs. Speed.
Pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

and Figure 2.10 for number of data byte transmissions per data packet delivered. As speed increases, the data load for both Flood and DREAM remains constant due to the flooding behavior that occurs in each protocol.

Since both LAR and DSR unicast data packets, both LAR and DSR have similar data loads (see Figures 2.9 and 2.10). Lastly, since fewer data packets are delivered at higher speeds (see Figure 2.5), both LAR and DSR have slightly higher data load for each data packet delivered at higher speeds.

2.4 Related Work

Previous simulation results have been presented for the protocols evaluated in this chapter. However, this is the first research to provide a detailed, quantitative evaluation comparing their relative performance. In addition, none of the results previously reported on LAR and DREAM used a simulation of a complete physical layer and MAC. We implemented both LAR and DREAM in NS-2 to provide our performance investigation with a complete physical layer and MAC.

Prior Results on DSR The results presented in [8] on DSR-P are quite different from the results presented herein. For example, all the data packet delivery ratios presented in [8] for DSR are over 95%. Their results are not comparable to ours because of the differences in our simulation environments. First, the average number of neighbors in [8] is much larger than our average number of neighbors (see Table 2.4). Second, the maximum average speed considered in [8] is only 10 m/s; our maximum average speed is 20 m/s. Third, the transmission range is 250m in a 1500x300m simulation area. Thus, the percentage of the simulation area that is covered by the transmission range is 43.6% of the simulation area. In our simulations, the percentage of the simulation area that is covered by the transmission range is only 17.4%. Lastly, the metric used for the x-axes in [8] is pause time, rather than speed. As discussed in Section 2.2, speed has a much greater impact than pause time on link

breakage rates [4].

The results presented in [32] on DSR-NP are also quite different from ours. The results in [32] evaluate pause times equivalent to 1 s; we evaluate much longer pause times uniformly chosen from 10 s $\pm$ 10%. In addition, only 15 CBR sources transmit data in [32] while 20 CBR sources transmit data herein. Lastly, results presented in [32] are taken from only 250 seconds of simulation time. As shown in Figure 2.4, there is high variability in the average number of neighbors during the initial seconds of simulation time for MNs using the random way-point model. Since the authors of [32] do not present confidence intervals for the unknown mean in the random scenarios, the precision of their estimates can not be determined.

Prior Results on LAR Many of the performance results presented in [36] have an x-axis for the transmission range, number of MNs, or the error factor in GPS location information (which is no longer relevant). Of the results in [36] that have an x-axis of speed, only one figure has a y-axis representing a statistic that we calculate. Although the x-axis in [36] defines speed as units per second instead of meters per second, and although a complete physical layer and MAC is not simulated in [36], the results presented in Figure 6(a) in [36] are comparable to the results presented in Figure 2.7 herein. Specifically, as speed increases, the number of control packet transmissions per data packet delivered increases for both LAR protocols and increases more substantially for DSR-NP's method of flooding route request packets.

Prior Results on DREAM The results presented in [2] are substantially different from ours. Specifically, the data packet delivery ratios presented for DREAM are all over 80% and the end-to-end delay presented for DREAM is smaller than the end-to-end delay presented for DSR. Since the simulation environment given in [2] is in clock ticks and units, it is difficult to compare their simulation environment with our simulation environment (which is in meters and seconds). There are, however, a few major differences in the simulation environments studied that are certain.

First, the results presented in [2] were obtained from the Maisie simulation package which does not offer a complete physical layer and MAC. Second, the mobility model used in [2] is a Brownian motion mobility model which creates a more stable network than the random way-point model [9]. Third, in [2], the transmission range is 40 units over a 100x100 unit simulation area. Thus, the percentage of the simulation area that is covered by the transmission range is 50.2% of the simulation area; in other words, little routing occurs in their simulation results. (As mentioned, the percentage of the simulation area that is covered by the transmission range in our simulations is only 17.4%.) Lastly, although we do not know what a clock tick is compared to a second, the 30 MNs in [2] only transmit (approximately) 1.5 to 12 packets per 300 clock ticks which we believe is a much smaller data load than the data load studied herein.

Prior Results on DREAM and LAR The only prior comparison (of which we are aware) of DREAM and LAR exists in [19]. Although only three figures comparing DREAM and LAR are given in [19], only one of the three figures exist herein. The authors of [19] found, as we have (see Figure 2.5), that DREAM is more robust to mobility than LAR. They attribute this fact to the partial flooding of data packets that occurs in the (cone) forwarding zone; we suspect, however, that this fact is due to the flooding of data packets that occurs from the recovery procedure. The packet delivery ratios presented in [19] for LAR are similar to the packet delivery ratios presented in Figure 2.5. The packet delivery ratios presented in [19] for DREAM, however, are much larger than the packet delivery ratios presented herein. We attribute this difference to not having contention and congestion fully modeled in [19].

2.5 Conclusions

Conclusion 1: *The added protocol complexity of DREAM does not appear to provide benefits over Flood.* In the DREAM protocol, each data packet is first flooded in the forwarding zone and then (possibly) flooded in the entire network through the recovery procedure. As discussed in Figure 2.6, the DREAM recovery procedure is used almost all the time if the MNs move; thus, the end-to-end delay of DREAM is much higher than the end-to-end delay of Flood. As illustrated in Figure 2.5,

compared to Flood, the DREAM protocol has equivalent data packet delivery ratio for all speeds. In addition, Figures 2.7 through 2.10 illustrate that the packet and byte network load of DREAM is comparable to the packet and byte network load of Flood. Thus, there appears to be no reason to include the additional protocol complexity of DREAM over the simple protocol of Flood.

Conclusion 2: *Location information improves DSR, especially at high speeds.* Figure 2.5 illustrates that using location information improves the data packet delivery ratio of DSR significantly. In fact, since LAR-NP offers higher data packet delivery ratio than DSR-P, *the use of location information is more beneficial than the use of promiscuous mode operation.* There is a cost for this increase in data packet delivery ratio. Specifically, at 20 m/s, the two LAR protocols improve the data packet delivery ratio of DSR-P by approximately 40% and the end-to-end delay by approximately 20%. The cost for this improvement is a 15% increase in the number of packet (control and data) transmissions for each data packet delivered and a 70% increase in the number of byte (control and data) transmissions for each data packet delivered.

The performance benefits are more substantial for DSR-NP. At 20 m/s, the two LAR protocols improve the data packet delivery ratio of DSR-NP by approximately 130% and the end-to-end delay by approximately 35%. In addition, the two LAR protocols decrease the number of packet (control and data) transmissions for each data packet delivered in DSR-NP by 30%. The cost for this improvement is a 15%

increase in the number of byte (control and data) transmissions for each data packet delivered. Since the cost of transmitting packets in a wireless network is much more severe than the cost of transmitting bytes, the increase in data packet delivery ratio is worth the extra overhead to include location information in DSR.

Conclusion 3: *Promiscuous mode operation improves the performance of DSR significantly.* As shown in Figure 2.5, the data packet delivery ratio for DSR-P at speeds greater than 1 m/s is significantly higher than that of DSR-NP. As shown in Figures 2.7 and 2.8, the control overhead for DSR-NP at speeds greater than 1 m/s is significantly higher than the control overhead for DSR-P. Thus, promiscuous mode operation improves the performance of DSR significantly.

Conclusion 4: *Our implementation of DREAM provides a simple location service.* Recently, a few of the location based routing protocols proposed (e.g., [31, 35]) have assumed the availability of some location service (e.g., Grid's Location Service [37]) to translate an MN's address to the MN's geographical location. The authors of DREAM proposed that an MN transmits location information adaptively based on when the MN has moved a specified distance from its last update location. Details on this location service, however, are not provided in [2]. In Section 2.1.3, we propose a solution for the transmission of location information adaptively.

Conclusion 5: *There is a tradeoff between average end-to-end delay and data packet delivery ratio.* We were able to achieve (almost) 100% data packet deliv-

ery ratio for LAR at high speeds when an infinite queue of data packets is allowed. In other words, data packets are stored in a queue until a route to the destination is found. Once a route becomes available, all packets in the queue for the destination are immediately transmitted. (Although we did not test it, an infinite queue in DSR should perform in a similar manner.) If data packet delivery ratio was the only important performance metric, we would set the time to hold packets awaiting routes to infinity. In this situation, the data packet delivery ratio would be maximized at the cost of average end-to-end delay.

Chapter 3

MOBILITY METRIC ALTERNATIVES

MANET protocols have been extensively studied and simulated in the past few years. Several comparative studies ([8], [19], [32] and [51]) have shown that there is no single protocol which works well in a wide variety of network conditions. A truly effective MANET protocol will combine the strengths of the best existing protocols while avoiding their weaknesses. An adaptive scheme that responds to the current network dynamics at each node shows promise in achieving this goal.

We use the term network dynamics to refer to the wide range of communication conditions a node can experience in a mobile ad hoc network. It refers to changing network topology, data congestion, shared medium contention, varying traffic patterns, varying traffic distributions, etc. For instance, node movement creates network topology changes via link breaks and link additions. These link changes require network protocol responses to ensure data services continue. Protocol responses in turn alter the traffic distribution in the network which also varies the congestion each node experiences. All these complex interactions result in certain network dynamics which are experienced differently by the nodes in the network.

A mobility metric quantifies the effect of node movement. It can then be used to

provide feedback by signaling the communication potential of the network. Such a mobility metric should be able to accurately indicate protocol performance, be protocol independent, and be obtainable by network nodes in real protocol implementations (i.e., it should not be dependent upon simulation artifacts such as mobility model parameters or movement patterns).

In this chapter we first present our specific requirements of a mobility metric. Next, we give a brief discussion of several alternative metrics used in the past along with some of their strengths and weaknesses, and present link duration as a mobility metric that satisfies our requirements and accurately indicates protocol performance. Finally, we present a set of simulation parameters to be used for all subsequent simulations based on the results presented in this chapter and Chapter 2.

3.1 Metric Requirements

Enabling adaptive protocols that can perform effectively throughout the demanding range of network dynamics requires a mobility metric that can accurately capture the networking challenges and effectively indicate the resulting performance of MANET protocols. Such a metric could provide feedback and allow the adaptation of a single protocol. Or the metric could facilitate a gradual change from one protocol to another in order to use the protocol that performs best in the current network scenario.

Past MANET simulation results have typically been reported by plotting protocol performance versus some parameter derived from the simulation mobility model being used. For instance, some research reports data packet delivery ratio or some other protocol metric (delay, overhead, etc.) as node pause time increases (see [8] and [51]). Mobility model input parameters are artificial quantities used as input into simulation models and are not suitable when MANET protocols are applied to real world networks. A mobility metric which is used as a MANET protocol feedback mechanism must be applicable to real networks with real nodes.

Our first step in a search for a mobility metric is to define a set of requirements that we feel must be met in order to enable adaptive MANET protocols.

1. **Computable in a distributed environment** *without* **global network knowledge.**

   Any metric which requires information from other network nodes and aggregates that information places an added demand on the network. This demand is non-trivial, and in some cases could require communication of mobility values from all, and to all, network nodes.

2. **A good indicator of protocol performance.**

   - **data packet delivery ratio**
   - **end-to-end delay**
   - **protocol overhead**

A mobility metric must be able to indicate or predict the protocol's performance. For example, as the mobility metric's value changes, this would indicate a corresponding change in end-to-end delay. When the node receives feedback via the mobility metric it can proactively adjust the protocol to keep the delay within the required application bounds.

3. **Feasible to compute (in terms of node resources).**

   Limited energy (as a result of battery power), limited processing ability, and limited memory availability are all resource constraints that mobile nodes might have. A mobility metric must be feasible to compute by each node participating in the network. For example, a mobility metric which requires frequent communication may not be feasible to compute.

4. **Independent of any specific protocol.**

   Any feedback that is dependent on a specific protocol requires that protocol to be present. A general purpose mobility metric, which can be used by any protocol as a feedback mechanism for adaptive operation, is preferred. Furthermore, a protocol independent mobility metric is applicable to many network abstraction layers. The mobility metric can enable adaptive operation of a MAC protocol as effectively as a routing or reliable transport layer protocol.

5. **Computable in real network implementations.**

   A mobility metric must be available to real network nodes in real network

implementations. As such, no simulation specific parameters or artifacts can be used. For example, actual node movements in real world networks will not always obey a previously studied pattern or model. The same is true for traffic model parameters or distributions.

3.2 Metric Alternatives and Related Work

While searching for a mobility metric to meet our requirements, several alternatives from the literature and elsewhere were considered. These alternatives are discussed in this section.

3.2.1 Mobility Model Parameters: Node Speed or Pause Time

Most research presents results according to node speed or some type of mobility model parameter such as pause time for the random way-point model. Using mobility model input parameters as a mobility metric links the adaptability of protocols to an artificial input that may not be present in a real network. As a result, it is apparent that using node speed and pause time fail requirement 5.

3.2.2 Information Unique to a Protocol

Some protocols have an inherent feedback mechanism. For example, the Dynamic Source Routing (DSR) protocol [8] could adapt the route request timeout based on the end-to-end delay of its route requests. In DSR, the overall end-to-end delay

could be reduced by adjusting the route request timeout, especially when network dynamics are high and frequent route requests are demanded. Enabling adaptivity in MANET protocols in this manner would require a custom feedback metric that is specific to each protocol, and possibly specific to each protocol parameter. This would require a great deal of work to identify, optimize, and show that the chosen unique protocol parameter was a good indicator of performance. In summary, using information unique to a protocol as the mobility metric fails requirement 4.

3.2.3 Minimal Route Change Metrics

In [28] the authors' goal for their mobility metric is to "evaluate the relative difficulty of routing in a given ad hoc network scenario." As such, they define two metrics that count the route changes between nodes. These types of mobility metrics fail two of our requirements. First, requirement 1 is not met since a network node can only know about multi-hop route changes with the assistance of and communication with intermediate nodes supporting the route. Second, while this notification is built into many MANET routing protocols, using this knowledge ties the mobility metric to a class of protocols and therefore does not meet requirement 4.

3.2.4 Average Relative Speed Between All Nodes

In [32] the authors present a mobility metric based on the average relative speed between nodes. While this does address the problem of two fast moving nodes which

are able to maintain a stable link because they are traveling in similar directions, it violates requirement 1 by requiring speed information from other network nodes. The mobility metric in [32] (average relative speed) is compared to the number of link changes experienced by a mobile node. This comparison is presented as support for the metric's ability to capture one aspect of node mobility.

3.2.5 Link Change Rate

The primary difficulty in mobile ad hoc networks is created not by node mobility, but by the motion of nodes relative to one another that forces the network topology, or node interconnections, to change. This notion led us to initially focus on the link change rate experienced by a node as a mobility metric [4].

Figure 3.1 shows the data packet delivery ratio achieved by a MANET routing protocol (LAR Box) as the link change rate increases[1]. Data packet delivery ratio appears to decrease as the number of link changes each node experiences increases. For example, delivery ratios for link change rate near 1000 range from 89% to 96%. Delivery ratios for link change rate near 2000 range from 82% to 92%. When two networks have similar link change rates, the one whose nodes tend to move more slowly will often have a substantially better data packet delivery ratio. When a linear fit is applied to the data in Figure 3.1, it demonstrates a coefficient of determination

[1]See Table 2.4 in Section 2.2 for simulation environment details.

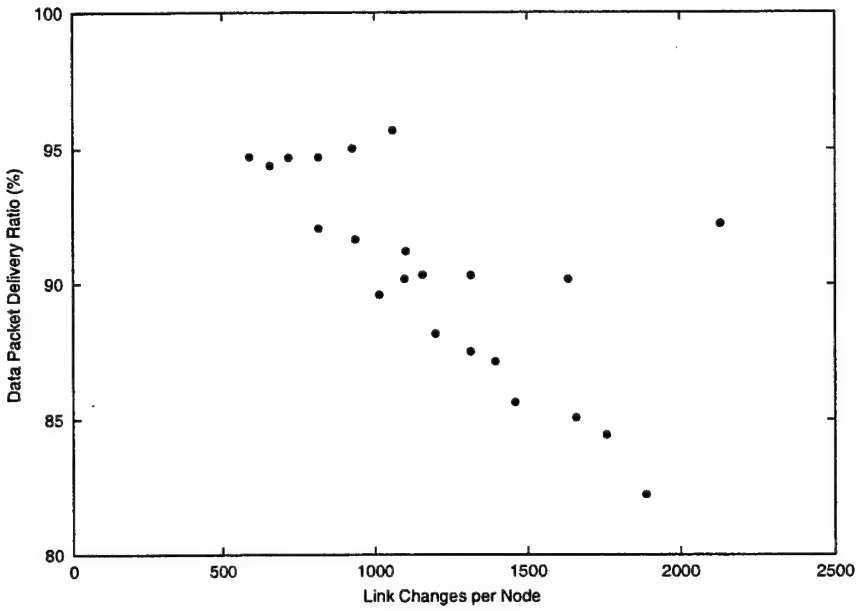

FIG. 3.1. Data Packet Delivery Ratio vs. Link Change Rate.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

just below 50%[2].

Figures for end-to-end delay (see Figure 3.2) and protocol overhead (see Figure 3.3) show similar results. Delay and overhead both increase with increasing link change rate, but there exist "tails" in the plots that are similar to those seen in Figure 3.1 and correspond directly to differing node speeds. The figures demonstrate a coefficient of determination when a linear function is fit of 24% for end-to-end delay

[2]We use the coefficient of determination as a quantitative measure to evaluate the predictive value of a given model. In this case it measures how well the link change rate can predict protocol performance.

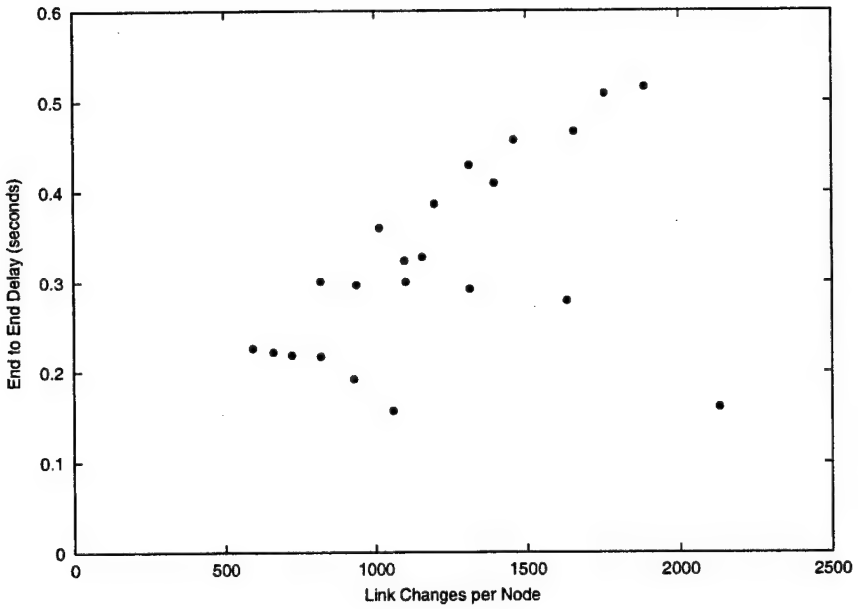

FIG. 3.2. End-To-End Delay vs. Link Change Rate.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

and 56% for overhead.

3.3 Link Duration

As shown in Figures 3.1, 3.2, and 3.3, link change rate does not reliably indicate the performance of a protocol in this set of simulations. In other words, link change rate fails requirement 2. In our analysis we realized that link change rate does not capture the longevity of communication links. Longer lasting links create more

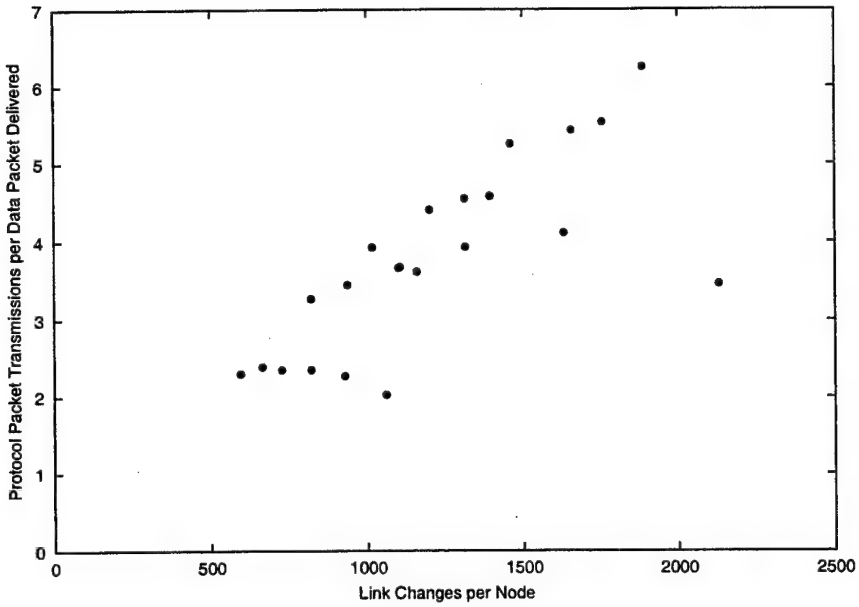

FIG. 3.3. Protocol Packet Transmissions per Data Packet Delivered vs. Link Change Rate.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

network stability, while shorter duration links create less network stability. An average link duration metric accurately captures this effect. It combines the link change rate and weights the changes by their stability (measured as number of seconds in duration).

For the simulation results presented in this section, the duration of one link is calculated as the time that two nodes are within transmission range of one another. Average link duration is then calculated on a per node basis by averaging the individ-

ual link durations experienced with all neighbors. The link duration presented on the x-axes of Figures 3.4, 3.5, and 3.6, is then the global network average of each individual node's link durations. In our simulation scenarios, all mobile nodes had similar movement patterns, and therefore, the variation of link duration between nodes was typically less than ten percent.

In a real implementation, neither global knowledge, nor an infinite window size (the time duration used to average the link connections) will be available to calculate link duration. Thus, our feedback agent focuses on calculating link duration in a real network (see Chapter 5).

3.3.1 Link Duration as an Indicator of Protocol Performance

Figure 3.4 plots the data packet delivery ratio as the link duration increases[3]. The figure shows that longer lived links create a more stable network, which in turn allows for a higher delivery ratio of data packets. In fact, as long as communication links exist on average for longer than 15-20 seconds, over 90% of the transmitted packets are delivered successfully. The fitted curve has a coefficient of determination of over 97%.

Figure 3.5 plots the end-to-end delay as average link duration increases. Again we see a strong relationship between link duration and delay. Longer lived links result in shorter end-to-end delays. As in Figure 3.4, we see that average link durations greater

[3]See Table 2.4 in Section 2.2 for simulation environment details.

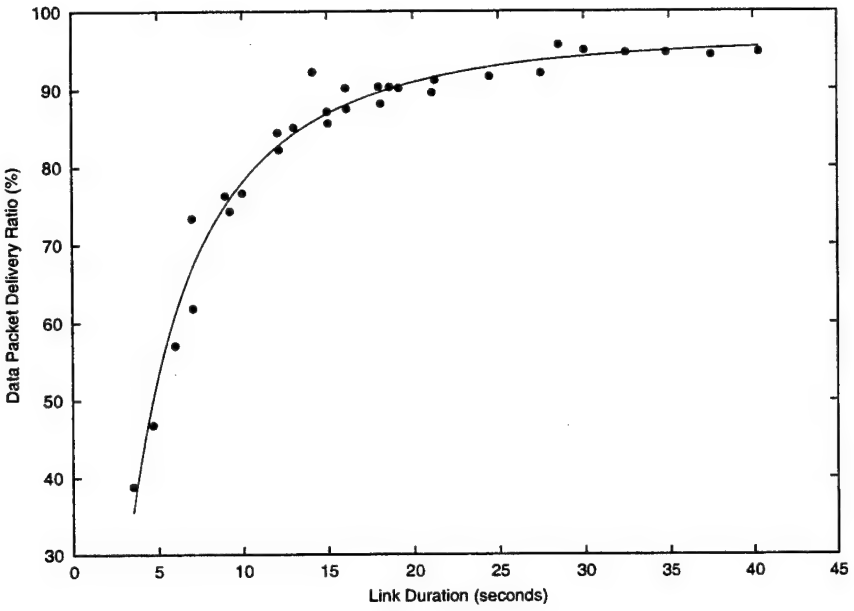

FIG. 3.4. Data Packet Delivery Ratio vs. Link Duration.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

than approximately 15 seconds result in low delay, and the curve nearly "flattens" out at higher link durations. The fitted curve has a coefficient of determination of 96.7%.

Lastly, Figure 3.6 plots the protocol overhead as average link duration increases. Once again we see a smooth relationship between link duration and protocol performance. Specifically, longer lived links result in lower protocol overhead. As in Figures 3.4 and 3.5, protocol overhead is at its lowest when average link durations are longer

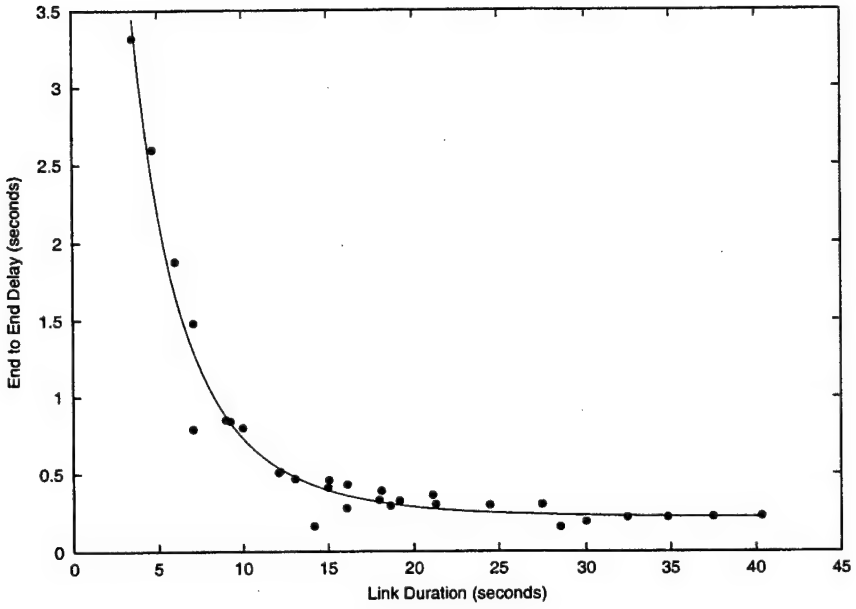

FIG. 3.5. End-to-End Delay vs. Link Duration.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

than 15 seconds. The fitted curve has a coefficient of determination of 98.5%.

3.3.2 Link Duration Meets Our Requirements

Link duration satisfies our five requirements for a mobility metric:

1. **Computable in a distributed environment without global network knowledge.**

   The only information needed by a network node to calculate link duration is

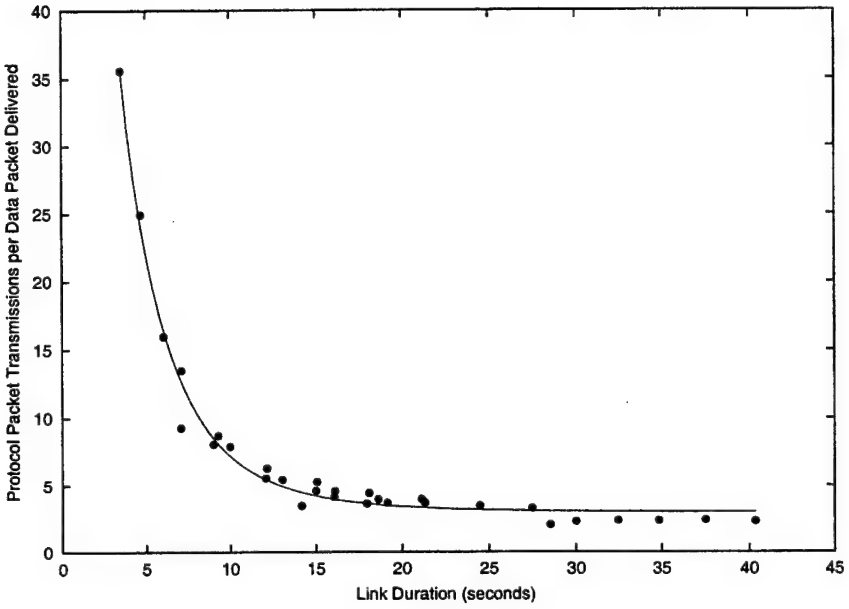

FIG. 3.6. Overhead vs. Link Duration.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

knowledge of its local changes in link status.

2. **A good indicator of protocol performance.**

   Figures 3.4, 3.5, and 3.6, demonstrate that link duration is a good indicator of data packet delivery ratio, end-to-end delay, and protocol overhead.

3. **Feasible to compute (in terms of node resources).**

   The efficient gathering of link durations is possible (see Chapter 5).

4. **Independent of any specific protocol.**

Measuring the duration of communication links is by its very nature independent of any protocol.

5. **Computable in real network implementations.**

   Link duration makes no reference to any artificial parameters. It measures a quantity that exists in all networks.

3.4 Simulation Parameters

The goal of simulating network protocols is to test their effectiveness in a wide range of network conditions. This section contains the specific simulation parameters which are used for all research presented in this dissertation, unless stated otherwise. The discussion of parameters in this section corresponds to the discussion of parameters in Section 2.2. In fact, except for extended speed and pause times, the parameters presented herein match the parameters used in Chapters 2 and 3.

Table 3.1. Input Parameters

Simulation time	1000 seconds Note: See Figure 2.4 for a discussion on our simulation duration. Specifically, the simulation duration is 2000 seconds; data, however, is only transmitted during the final 1000 seconds.
Simulation area	300x600m
Number of MNs	50
Transmission range	100m

Table 3.2. Derived Parameters

Node density	1 node per 3,600 m^2
Coverage area	31,416 m^2
Transmission footprint	17.45%
Maximum path length	671m
Network diameter (max. hops)	6.71 hops
Average neighbors	8.73 (no edge affect)
Average neighbors	7.76 (edge affect)

The simulation input parameters are listed in Table 3.1. Derived parameters (see Table 3.2) are calculated directly from the input parameters [4]. Node density is simply the number of nodes divided by the total simulation area. Coverage area is the area of the circle whose radius is the transmission distance. The transmission footprint of a node is the percentage of the simulation area covered by a node's transmission. It is derived from the transmission range of the node and the size of the simulation area. The maximum path length is the distance from the lower left corner to the upper right corner in the simulation area. The network diameter is the maximum path length divided by the transmission range. Finally, the average number of neighbors indicates the network connectivity. The value labeled "no edge affect" is calculated by dividing the coverage area by the node density. The value labeled "edge affect" takes into account the fact that nodes near the edges do not have neighbors on all sides of the node.

We use the random way-point mobility model with the average speed and pause

Table 3.3. Mobility Model

Mobility model	random way-point
Mobility model parameters	speed = [5, 10, 20, 30, 40] m/s pause time = [0, 10, 20, 30, 40, 50] seconds

Table 3.4. Data Traffic Model

Traffic type	Constant Bit Rate (CBR)
Number of senders	20
Number of receivers	20
Data payload	64 bytes
Packet rate	4 packets per second
Link bandwidth	2 Mbps
Traffic pattern	communicating pairs (peer-to-peer)

times shown in Table 3.3. The data traffic model and parameters used are included in Table 3.4. Finally, we simulate all protocols in NS-2 with the specific parameters shown in Table 3.5.

In all our research presented, we gather a standard set of performance results. The standard metrics gathered are:

- data packet delivery ratio,
- end-to-end delay,
- protocol overhead (packets and bytes per data packet delivered),
- data overhead (packets and bytes per data packet delivered),

Table 3.5: Simulator

Simulator Used	NS-2 (version 2.1b7)
Medium Access Protocol	IEEE 802.11
Number of Trials	10 minimum, 20 on some cases
Confidence Interval	95%

- total overhead (packets and bytes per data packet delivered), and
- average route length (number of hops) per data packet delivered.

We note that all overhead values (packets and bytes) are divided by the number of data packets or bytes actually delivered. In other words, we normalize results to ensure that the overhead values won't appear to improve as a result of the protocol performance degrading. Another important metric is the average route length. It is essential that route lengths are long enough to ensure meaningful conclusions can be drawn about routing performance and effectiveness.

3.5 Conclusions

A mobility metric can be used to enable MANET protocols to adapt. We have enumerated our requirements for a mobility metric (see Section 3.1), and then discussed several alternative metrics and their adherence to these requirements (see Section 3.2). In general, previously proposed mobility metrics either do not provide good indicators of protocol performance, or require global data from other nodes to

be calculated. One metric, link duration, has been shown to satisfy our mobility metric requirements. In Chapter 5, we discuss our development of a feedback agent that quantifies network dynamics via link duration.

In Section 3.4 we discuss the simulation parameters and performance metrics we use in Chapters 4 - 7. All performance results in these chapters are presented with the average link duration on the x-axis.

Chapter 4

PROVIDING LOCATION INFORMATION

Chapter 2 summarizes the results presented in [11], which show that the use of location information in an ad hoc network improves routing performance of unicast communication. As a result, a number of ad hoc network unicast routing protocols use location information, e.g., LAR [36], DREAM [2], GPSR [35], and GRA [31]. Each protocol makes different assumptions about the availability of location information. For instance, LAR and DREAM include the dissemination of location information as a part of the protocol, while GPSR relies on the presence of an outside location service to provide location information to all the nodes. The use of location information in these and other protocols led us to develop and evaluate the performance of three location services: the Simple Location Service (SLS), the DREAM Location Service (DLS), and the Reactive Location Service (RLS).

The details of these location services are specifically designed to provide a broad coverage of the spectrum of non-hierarchical location services. Two of the services are proactive (SLS and DLS), and one is reactive (RLS). In addition, the two proactive services represent two primary ways to distribute information in a network:

1. through the exchange of table information between neighbors, similar to the

well known wired routing protocol RIP (Routing Information Protocol) [26], and

2. through the exchange of location information between all nodes in the network, similar to the well known wired routing protocol OSPF (Open Shortest Path First) [40].

RLS is a reactive location service that queries location information on an as needed basis, i.e., in a manner similar to LAR [36] or DSR [34]. Each node in RLS maintains a location table; entries in the location table of a node are purged periodically based on the age of the location information. Complete details of each protocol are included in Sections 4.2, 4.3, and 4.4.

4.1 Data Structures and Techniques

Each mobile node maintains a location table in order to have location information on other nodes in the network available. This table contains an entry on every node in the network whose location information is known, including the node's own location information. A table entry contains node identification, the coordinates of the node's location based on some reference system, the current speed of the node, and the time this location information was obtained.

When a location request occurs, a node first looks in its location table for the information. If the information is not available in the table, the node floods a location

request packet to all nodes in the ad hoc network. A reply to this location request packet is transmitted by the node whose location was requested; nodes that hear the reply to the location request update their table in a promiscuous manner. In other words, all three protocols include a flooding mechanism which is used (if necessary) to obtain a node's location information. The primary difference in the protocols is in how they maintain their location tables.

4.2 DREAM Location Service (DLS)

4.2.1 Protocol Description

We call this location service the DREAM Location Service (DLS) since it is similar to a location service proposed by the authors of DREAM [2]. Each location packet (LP), which updates location tables, contains the coordinates of the source node based on some reference system, the source node's speed, and the time the LP was transmitted. The transmission frequency and distance each location packet propagates uses the same technique as our optimization to DREAM (see Section 2.1.3).

DLS is similar to the Open Shortest Path First (OSPF) Internet standard [40]. OSPF is a link state routing algorithm developed for wired networks in which each network router periodically broadcasts (floods) the state of all directly connected links to the entire network. In the same manner, a DLS node periodically floods its

individual location information to the entire network. DLS, however, was developed for mobile networks; thus, it has a built in mechanism to change update frequency based on node speed. Furthermore, DLS distinguishes between nearby and faraway nodes; OSPF has no such distinction.

4.2.2 Implementation Decisions

For the transmission of nearby/faraway LPs, we set α to 4, X to 13, and Y to 23 seconds. (We optimized these three values via numerous simulation trials.) To avoid LPs being transmitted by neighboring nodes at the same time (and, thus, colliding), each mobile node offsets the transmission of their LPs with a random jitter. Lastly, if a location table entry in DLS is older than 46 seconds, the information in the entry is considered *outdated* and deleted. If a node receives a location request packet addressed to itself, then the node replies with an LP containing its location information.

4.3 Simple Location Service (SLS)

4.3.1 Protocol Description

A node using the Simple Location Service (SLS) transmits a location packet (LP) to its neighbors at a given frequency. The frequency a mobile node transmits LPs

adapts according to the node's rate of location change, via a similar procedure used for nearby nodes in DLS:

transmit LP: $(\frac{T_{range}}{\alpha}) * (\frac{1}{\nu}) = \frac{T_{range}}{\alpha\nu}$

or

one at least every Z seconds.

Each LP in SLS contains E entries from the node's location table; the E entries are chosen from the table in a round robin fashion. In other words, each LP transmission shares the location information of E other nodes in the ad hoc network with the node's neighbors. Since LPs are transmitted periodically, there is a good chance that all the location information a node knows will be shared with its neighbors.

A node using SLS periodically receives a location packet from each of its neighbors. The node updates its location table based on the received table entries, such that each location information with the most recent time (between the node's own table and the received table) is maintained.

There are some similarities of SLS to the Routing Information Protocol (RIP) [26], an Internet standard. For example, both transmit tables to neighbors on a periodic basis. Differences between these two protocols include the following: SLS sends partial location tables compared to the total routing tables sent by RIP and, unlike RIP, a node using SLS utilizes its current location table in the calculation of its new location table. Lastly, RIP was not developed for an ad hoc network environment;

thus, the frequency of RIP update packets is only based on time.

4.3.2 Implementation Decisions

For the transmission of location packets, we set α to 4, Z to 13, and E to 25. As in DLS, to avoid LPs being transmitted by neighboring nodes at the same time (and, thus, colliding), each mobile node offsets the transmission of their LPs with a random jitter. If a location table entry in SLS is older than 46 seconds, the information in the entry is considered *outdated* and deleted. If a node receives a location request packet addressed to itself, then the node replies with an LP containing up to E entries from the node's location table (including its own location). We discuss the effect of different values for E in Section 4.5.

4.4 Reactive Location Service (RLS)

4.4.1 Protocol Description

The Reactive Location Service (RLS) is an on demand location service. When a mobile node requires a location for another node, and the location information is either unknown or expired, RLS floods a location request packet through the network until it reaches the node whose location is requested. The node whose location is requested returns a location reply packet via the reverse source route obtained in

the location request packet. In other words, each location request packet carries the full route (a sequenced list of nodes) that a location reply packet should be able to traverse in its header. Since IEEE 802.11 requires bidirectional links in the delivery of all non-broadcast packets, we assume bidirectional links in RLS. (If bidirectional links are not available, this requirement can be removed via the manner proposed in DSR [8].)

Like DLS and SLS, RLS inherits some aspects of its functioning from existing protocols. Specifically, the location request/reply mechanism is essentially the route request/reply mechanism included in DSR and LAR (see [34] and [36] respectively). We apply this reactive behavior to the discovery of node locations in RLS. In addition, requesting desired information from neighbors first (also called ring zero search) and allowing intermediate nodes to reply to a request are two features from DSR that we make use of in RLS.

4.4.2 Implementation Decisions

A node using RLS removes *outdated* entries from its location table if the node in the entry is believed to have moved more than one transmission distance (100m) since the last location information was received from that node. An entry also becomes outdated in the same manner as SLS and DLS; specifically, if a location table entry in RLS is older than 46 seconds, the information in the entry is deleted. In addition, similar to SLS and DLS, each node offsets its transmission of location packets with

random jitter in order to avoid collisions. Lastly, the timeout for a one hop location information request is 30 ms.

4.5 Results

4.5.1 Simulation Environment

Each location service was implemented according to the above protocol descriptions in the network simulator NS-2 [21]. The performance of each location service was initially tested in a manner similar to Chapter 2 and [11]. That is, simulations were run and a complete set of performance results were presented in [10] with node pause time equal to 10 seconds and speed ranging in $[1, 5, 10, 15, 20]$. These results are presented with speed on the x-axis.

The results presented in this section compliment and extend the results presented in [10] by including and focusing on more demanding mobility scenarios. In addition, we present the results with respect to link duration. In other words, the simulation scenario's average link duration is shown on the x-axis in the included plots. Table 4.1 lists the random way-point mobility model node speeds which we use and their resulting link durations. The node pause time is zero for all speeds (see Section 4.7 for a discussion). All other simulation environment parameters are in accordance with the details outlined in Section 3.4.

Speed (m/s)	Link Duration (seconds)	Speed (mph)
40	3.5	90.0
35	4.0	78.8
30	4.7	67.5
25	5.6	56.3
20	7.1	45.0
18	7.9	40.5
16	8.8	36.0
14	10.1	31.5
12	11.8	27.0
10	14.1	22.5
9	15.8	20.3
8	17.7	18.0
7	20.3	15.8
6	23.7	13.5
5	28.5	11.3

Table 4.1. Node Speed and Link Duration. All Node Pause Times Are Zero.

4.5.2 Performance

We evaluate DLS, SLS, and RLS in both performance areas (e.g., the percentage of location requests that are answered) and overhead areas (e.g., the number of location packets transmitted per location request answered). All the performance results presented are an average of 10 different simulation trials. We calculate a 95% confidence interval for the unknown mean, and we plot these confidence intervals on the figures. Since most of the confidence intervals are quite small, we are convinced that our simulation results precisely represent the unknown mean.

The percentage of location requests that are answered versus link duration is

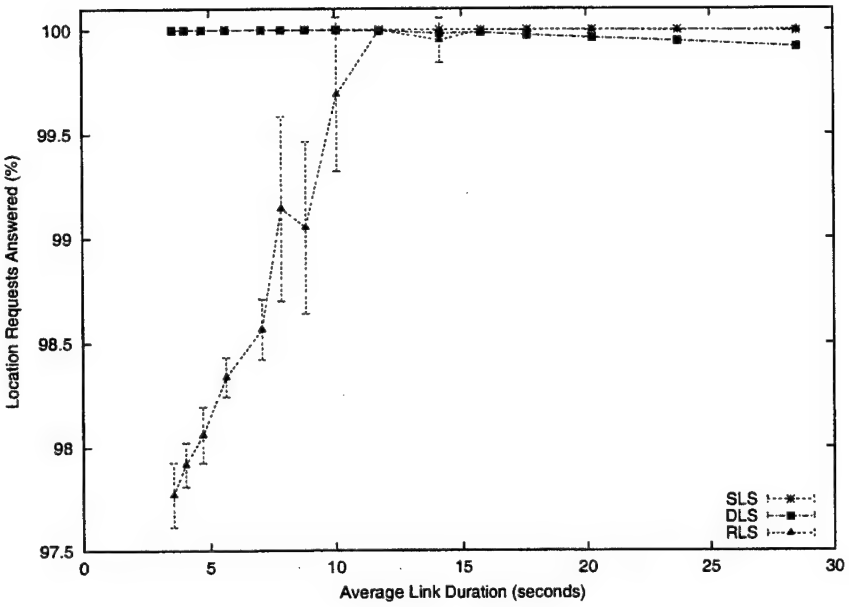

FIG. 4.1. Location Requests Answered vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

shown in Figure 4.1. We note that the results for all three protocols range between 97.5% and 100%. In other words, all three protocols provide location information on a given node almost all of the time. When link durations become short SLS and DLS continue to perform well, but RLS performance begins to drop. In other words, DLS and SLS perform better than RLS in high mobility (short link duration) situations due to network partitions (see the following discussion).

We note the larger confidence intervals for the RLS data points as link dura-

tions fall below 12 seconds. Examination of the mobility scenarios reveals that in scenarios with link durations longer than 12 seconds the network seldom partitions. In scenarios with link durations shorter than 12 seconds network partitioning occurs with increasing frequency. Since SLS and DLS retrieve location information from a local table, the performance of these two protocols are less sensitive to partitioning. RLS on the other hand is more seriously affected by network partitions, since RLS's reactive location request mechanism floods a location request packet in the network. The same increased confidence intervals are present for RLS in all the performance and overhead results.

While Figure 4.1 gives us an indication of available location information on a given node, it does not answer the question of how accurate the information is. Figure 4.2 plots the average location error (in meters) of the protocols versus link duration. The average location error is the actual location of the mobile node (at time t) minus the location of the mobile node provided by the location service (at time t). As shown, there is a prominent performance difference between the protocols in our mobility scenarios; overall, SLS is the most accurate, DLS the least accurate, and RLS performs between SLS and DLS. When mobility decreases (longer link durations which corresponds to slower moving nodes) DLS and RLS increase their accuracy while SLS remains nearly constant.

SLS provides the most accurate location information when the network is highly

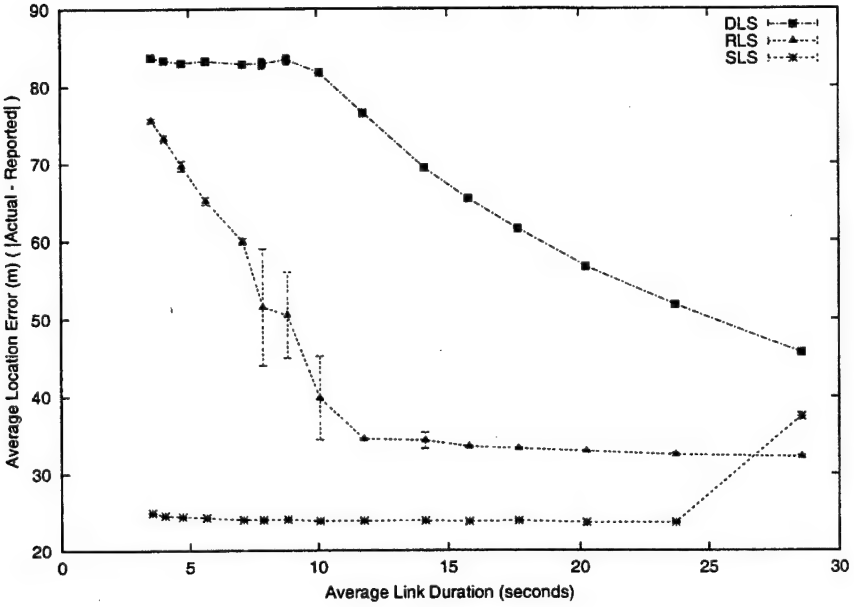

FIG. 4.2. Error of Location Responses vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

mobile (short link durations). SLS benefits from higher speeds, since a mobile node shares its location table entries with more nodes when the mobile node is moving quickly. Similar to SLS, a node using DLS transmits more location packets at higher speeds. Unlike SLS, however, the location packets (LPs) of DLS are flooded in the network. Due to contention issues from the flooding of LPs, the location information provided by DLS is less accurate. The accuracy of DLS improves as link duration increases since the amount of flooding decreases as speed decreases. Lastly, the lo-

cation error provided by RLS increases rapidly when link durations are less than 12 seconds due to the network partitioning problem discussed.

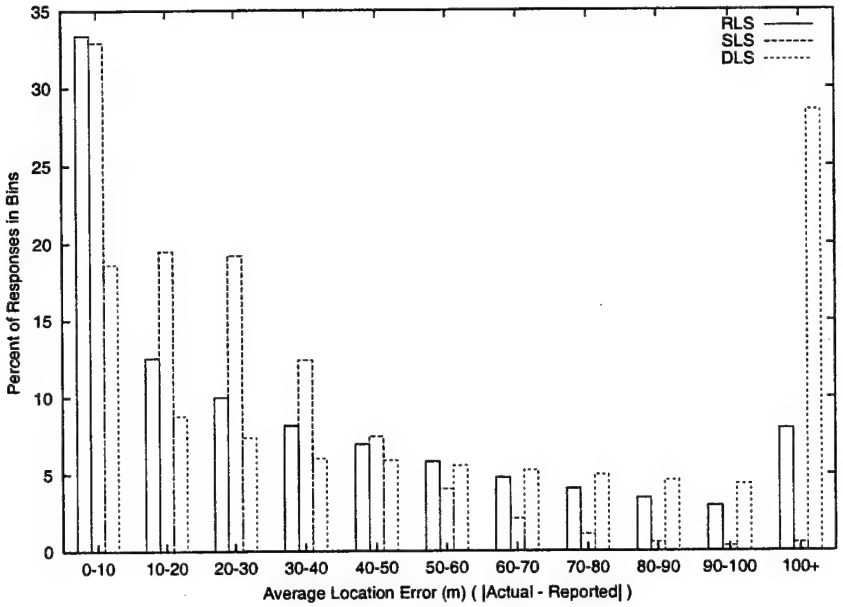

FIG. 4.3. Histogram of Location Error in Location Responses.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

We evaluate the location error of each protocol more closely in Figure 4.3. This figure gives a histogram of the location error provided by each protocol when link duration is 16 seconds (see Figure 4.2). We define a location information response *invalid* if the error on the location information is greater than the transmission range. Thus, the percentage of location errors that are invalid for each protocol is shown

in bin 100+. We note that over 25% of the location responses returned by DLS are invalid. Figure 4.3 also illustrates that RLS (unlike SLS) returns location responses that are *invalid* almost 10% of the time.

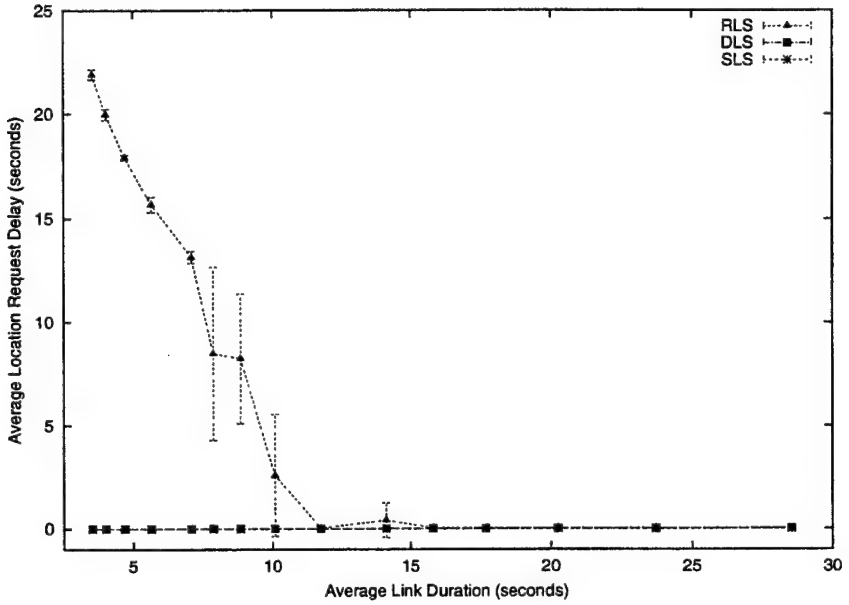

FIG. 4.4. End-to-End Delay for Location Request vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

Both Figure 4.4 and Figure 4.5 concern the amount of delay in obtaining a response to a location request. Figure 4.4 plots the average end-to-end delay of a response to a location request versus link duration. Figure 4.5 plots the percentage of location requests that are answered by the requesting node via its location table;

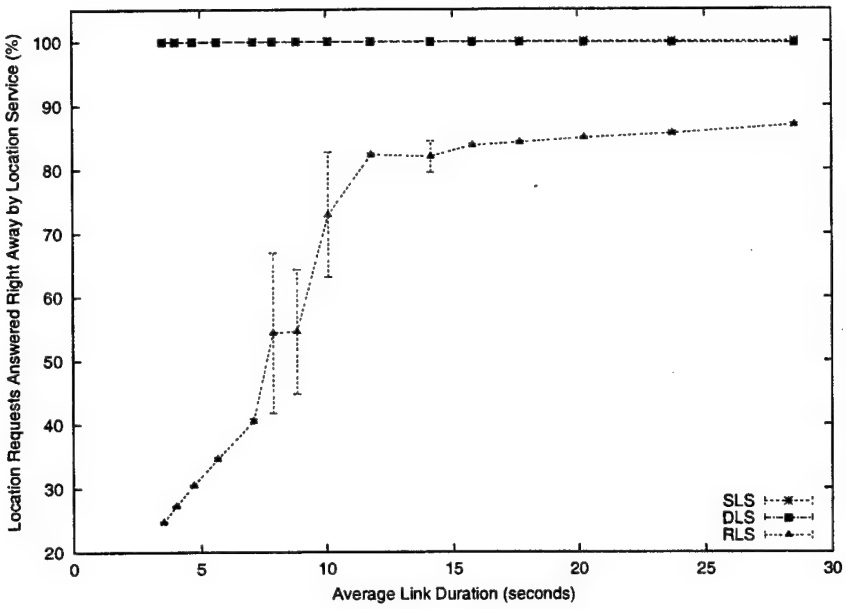

FIG. 4.5. Location Answers Available in Location Table vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

in other words, Figure 4.5 indicates the number of location table entries that exist in each of the protocols over the number of nodes in the network. As shown, the delay for the two proactive protocols (SLS and DLS) is lower than that of the reactive protocol (RLS) at short link durations (see Figure 4.4). In fact, the average delay for SLS is zero for all scenarios since it answers all location requests right away. The average delay for DLS is zero for many scenarios as well; in DLS, scenarios with long link durations are not able to answer location requests right away. When network

mobility increases (short link durations), RLS's reactive location requests cannot keep up with link changes and network partitioning. The significant increase in delay is a result of many location requests never being answered.

The proactive protocols (SLS and DLS) answer nearly every location request right away, that is, directly from their stored location tables (see Figure 4.5). As expected, RLS, a protocol that obtains location information on an as needed basis, has the lowest percentage of location requests answered right away at all link durations. Comparing Figures 4.4 and 4.5 emphasizes the point where RLS fails to answer location requests right away (less than 12 seconds).

4.5.3 Overhead

Figures 4.6, 4.7, and 4.8 illustrate the overhead that each location service requires. Figure 4.6 shows the number of location packet transmissions for each location request provided as link duration increases. This figure helps capture the power overhead requirements of each protocol. All three protocols have a flooding component. Flooding in DLS is one of the proactive tasks in the protocol. Flooding in SLS, on the other hand, only occurs when the requested location information is not available in the location table; thus, the number of packets transmitted for each location request answered in SLS is quite small (never greater than one). We compare DLS and SLS more closely below. RLS also floods only when the requested location information is not available in the location table; however, since RLS is not a proactive protocol,

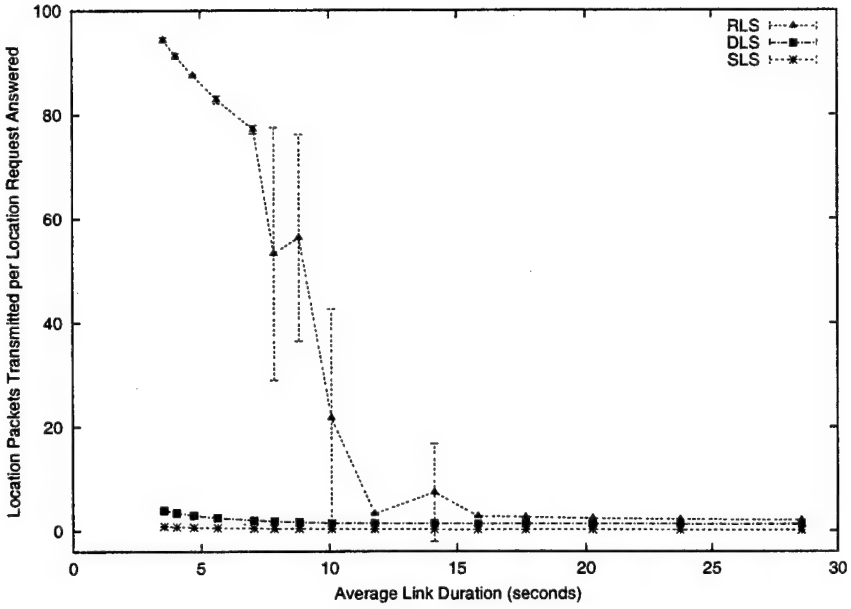

FIG. 4.6. Location Packet Overhead vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

RLS is more likely than SLS to flood location requests. Furthermore, this task is more likely to occur when mobility is high (short link durations). In fact, when link durations are shorter than 12 seconds, the packet overhead required for RLS quickly rises to greater than an order of magnitude more than DLS and SLS.

Figure 4.7 illustrates the number of location byte transmissions for each location request answered as link duration increases (note Figure 4.7 y-axis is logarithmic scale). This figure helps capture the bandwidth requirement of each protocol. The

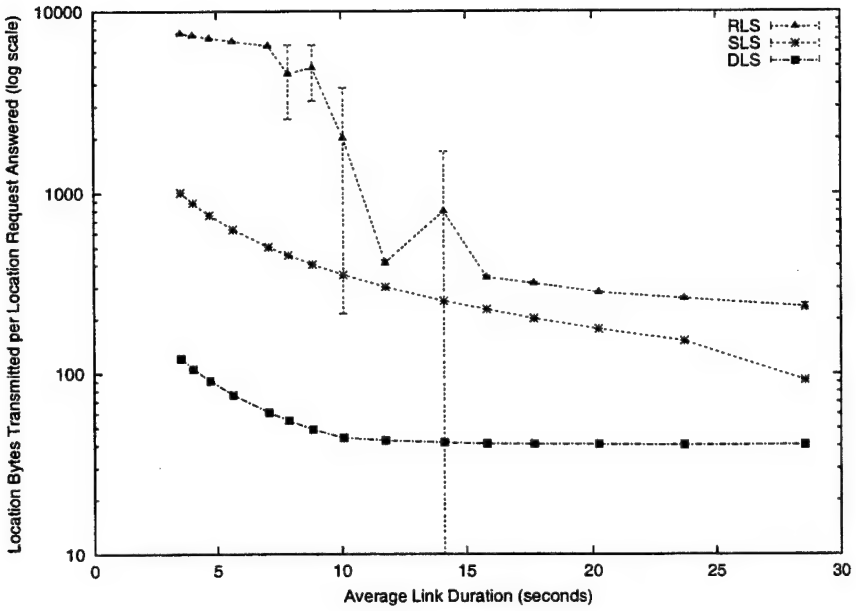

FIG. 4.7. Location Byte Overhead vs. Link Duration.
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

bandwidth requirement of RLS increases as network mobility increases, which corresponds to the increase in Figure 4.6. In other words, at low mobility, location information only expires due to the timer; at high mobility, location information expires more frequently, which generates more location requests and more overhead.

Finally, Figure 4.8 compares DLS and SLS packet overhead more closely as link duration changes. We zoom in on the performance of these two protocols; as a result RLS overhead only appears for long link durations. Figures 4.7 and 4.8 emphasize

one of the key differences between DLS and SLS. DLS transmits more packets, all of which are small. SLS on the other hand transmits fewer packets. However, each SLS overhead packet is several times larger than a DLS overhead packet. Higher byte overhead for SLS is acceptable if the performance of SLS is also higher. Figures 4.2 and 4.3 validate that the SLS error on location information is substantially less than the errors provided by DLS.

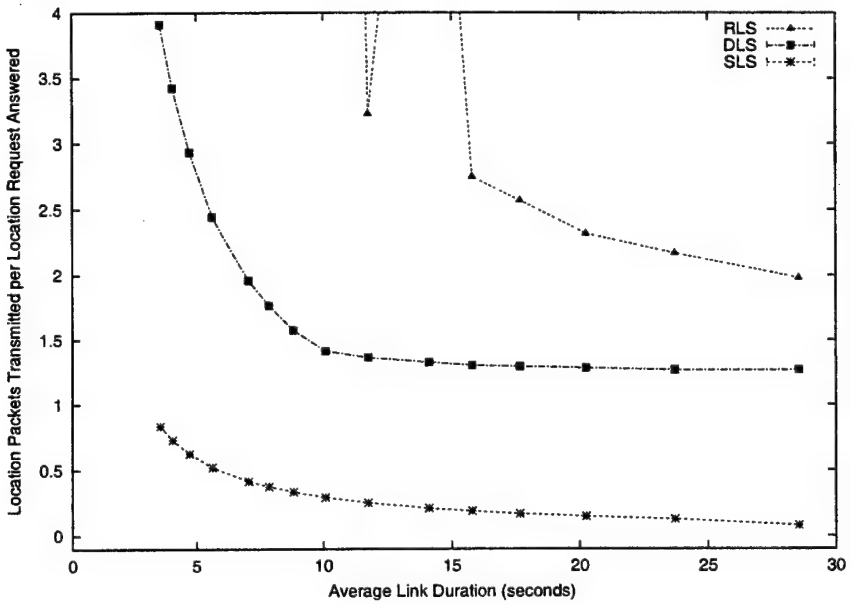

FIG. 4.8. Location Packet Overhead vs. Link Duration (Zoomed).
Speed = [5,6,7,8,9,10,12,14,16,18,20,25,30,35,40] m/s, pause time = 0 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

Overall, SLS offers higher performance (see Figures 4.2 and 4.3) and lower over-

head (see Figure 4.8) than RLS and DLS. Since SLS is preferred over both RLS and DLS, we evaluate the main input parameter associated with SLS. Specifically, a smaller (larger) SLS update table size should decrease (increase) both overhead and performance. We evaluate this trade off in the next section.

4.5.4 SLS Table Size Evaluation

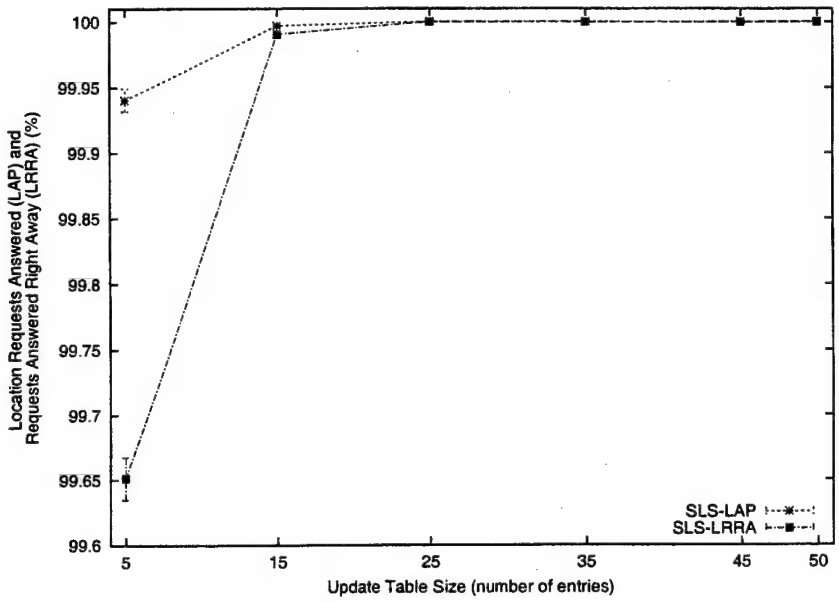

FIG. 4.9. Percent of Location Requests Answered and In Table vs. SLS LP Table Size.
Link duration = 16 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

In this section, we investigate a desired value for E, the number of location table

entries that are transmitted in each location packet, in a scenario with moderate mobility (average link duration is 16 seconds). Figure 4.9 combines the y-axes of Figure 4.1 and Figure 4.5 with an x-axis representing E. As shown, the performance of SLS increases as the number of location table entries increases in the location packets. In addition, since location information is often immediately available (i.e., in the node's cache), the average location request delay is always low. Specifically, the average location request delay decreases from 0.015 to 0.001 as the number of location table entries in a location packet increases from 5 to 25. Lastly, while Figure 4.9 illustrates that location information is (almost) always available, it does not answer the question of how valid the information is.

Figure 4.10 shows the accuracy of the location information provided by the mobile nodes. Again, average link duration is 16 seconds. Figure 4.10 plots the average location error (in meters) of SLS as the number of location table entries increases. As in Figure 4.2, the average location error is the actual location of the mobile node (at time t) minus the location of the mobile node provided by the location service (at time t). As expected, the average location error decreases as the number of location table entries increases. Furthermore, even when E is small, the average location error is competitive with the average location error provided by DLS at a link duration of 16 seconds; i.e., compare the results of Figure 4.10 with the results of Figure 4.2 at 16 seconds.

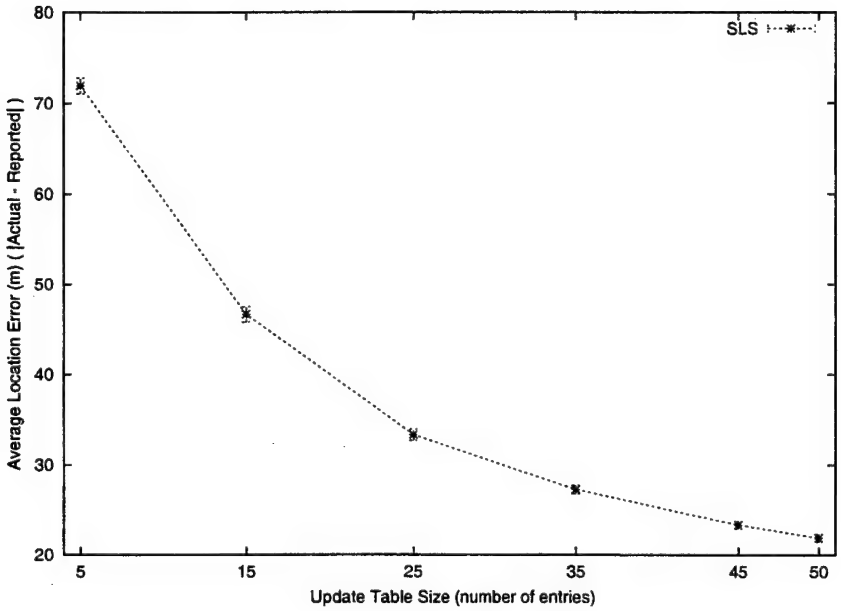

FIG. 4.10. Error of Location Responses vs. SLS LP Table Size.
Link duration = 16 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

Table 4.2 illustrates the overhead changes as the number of location table entries change in each location packet. The column labeled packets (bytes) is the number of location packet (byte) transmissions for each location request provided. As expected, the overhead, in terms of bytes, increases as the number of location table entries increases in each location packet. The overhead, in terms of packets, initially decreases (as one would expect) as the number of location table entries increases. The overhead, in terms of packets, remains constant as the number of location table entries increases

table size	packets	bytes
5	0.460	60.209
15	0.183	67.394
25	0.177	107.462
35	0.177	149.881
45	0.177	192.254
50	0.177	213.394

Table 4.2. SLS LP Table Size
Link duration = 16 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

from 25 to 50.

Based on the results in Table 4.2 and Figures 4.9 and 4.10, a desired number of location table entries in a location packet is between 15 and 25 for the simulation environment we evaluated. Increasing the number of location table entries higher than 25 increases overhead (in terms of bytes) with no associated increase in the percentage of location requests that are answered and little associated increase in the accuracy of location information provided. In our simulation environment 50 mobile nodes exist. Thus, a guideline for SLS is to set E to 30-50% of the mobile nodes in the ad hoc network.

4.6 Conclusions

In this chapter, we have proposed and evaluated (via simulation) three location services for an ad hoc network. One of the three protocols evaluated is a reactive

protocol. The other two protocols evaluated proactively transmit either location tables to neighbors or location information to everyone. An effective location service can be used to improve the performance and scalability of a routing protocol that requires location information (e.g., GPSR [35]).

Figures 4.2 and 4.3 illustrate that DLS, a proactive protocol that periodically floods location information, is unable to provide accurate location information (especially when mobility is high). On the other hand, a proactive protocol that periodically shares location table entries with neighbors (such as SLS) offers advantages over DLS in terms of simplicity, fewer packet transmissions (see Figures 4.6 and 4.8), and higher performance (see Figures 4.2 and 4.3).

When the proactive SLS protocol is compared with the reactive RLS protocol, we discover that the flooding requirements of RLS are much more costly in terms of both packets and bytes transmitted (see Figures 4.6 and 4.7). In addition, the percentage of *invalid* location responses provided by RLS is much higher than SLS (see Figure 4.3). We, therefore, conclude that our Simple Location Service is preferred over both our Reactive Location Service and DREAM's Location Service.

4.7 Simulation Parameter Discussion: Pause Time = 0

In Section 3.4, we discuss varying pause time between 0 and 50. In this chapter, however, we set pause time to zero. Figure 4.11 shows the average location error

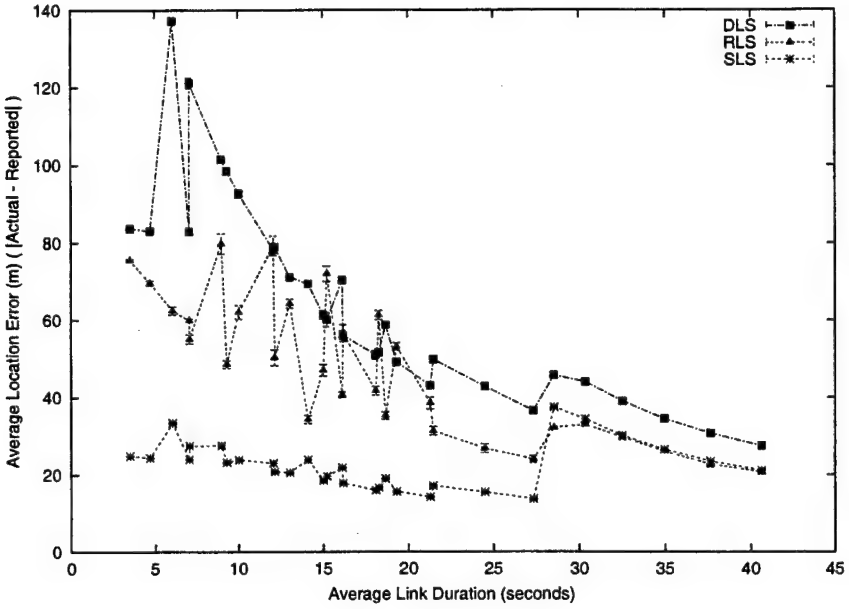

FIG. 4.11. Error of Location Responses vs. Link Duration.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

as link duration increases for all speeds in $[5, 10, 20, 30, 40]$ and all pause times in $[0, 10, 20, 30, 40, 50]$. When presented in this manner a clear trend is hard to identify. The erratic behavior of DLS and SLS is due primarily to their speed-based location packet update mechanisms. Both DLS and SLS adapt to mobility by sending location update packets more frequently as speed increases. The erratic behavior of RLS is due primarily to its speed-based location table update mechanism. RLS adapts to mobility by removing entries from its location table more frequently as speed increases. We

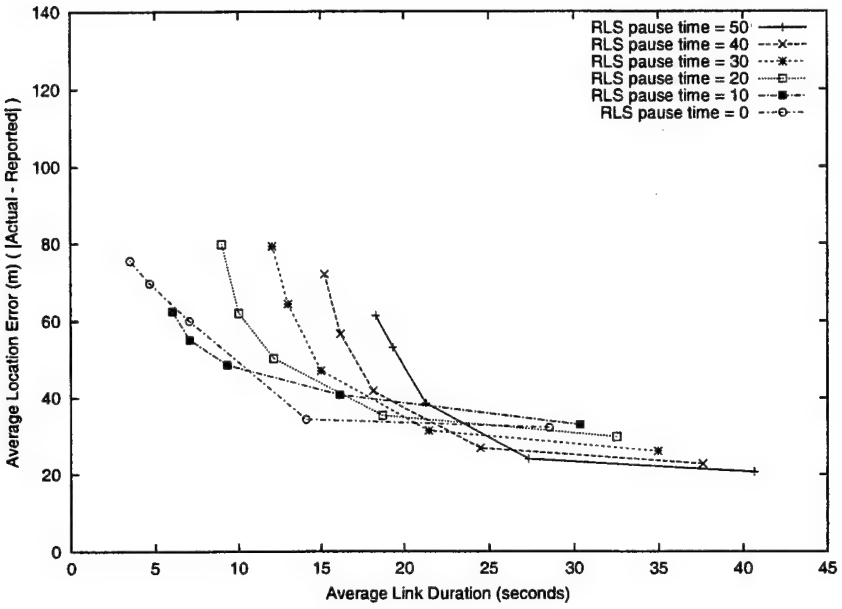

FIG. 4.12. Error of Location Responses vs. Link Duration - RLS only.
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
50 requesting nodes, 2 location requests per second.

have shown in Chapter 3 that speed is not always an accurate measure of mobility, and this is especially true when pause times are included. In fact, the erratic behavior in Figure 4.11 strengthens the argument against using only node speed as a mobility metric.

Figure 4.12 isolates RLS and shows the average location error for differing pause times. The data points included for each pause time represent different speeds. Speeds increase from left to right for each pause time line and range in $[5, 10, 20, 30, 40]$.

Figures 4.11 and 4.12 emphasize that speed-based adaptation affects performance and makes the interpretation of results by mobility (link duration) difficult. As a result, the performance and overhead results we include in Section 4.5 are for a wide variety of speeds (see Table 4.1) all with zero pause time. In other words, we do not obscure the performance trends as a result of the complex interaction between speed and pause time which the random way-point mobility model can introduce.

Chapter 5

MEASURING MOBILITY AND PROVIDING FEEDBACK

Mobile ad hoc networks face many challenges for successful communication. As we discuss in Chapter 1, the primary challenges are mobility, contention, and congestion. The latter two are applicable to both wireless networks and wired networks. The unique challenge which distinguishes mobile networks from all other networks is mobility. Thus, our primary focus for feedback to mobile nodes concerns the current *local* mobility characteristics of the network.

Chapter 3 reports the results in [5] which presents link duration as a good indicator of protocol performance. In this chapter, we discuss the development and evaluation of a generic feedback agent which resides on each mobile node, gathers the required link duration information, and provides the information to any protocol. We discuss the key issues surrounding the design, implementation, and evaluation of our mobility feedback agent in the following sections.

5.1 Requirements of a Mobility Feedback Agent

In this section we list three primary requirements of a mobility feedback agent. A feedback agent in an ad hoc network environment must incur minimal cost in terms of

node processing, power consumption, added network overhead, etc. Overloading the requirements for such a feedback agent would be in direct opposition to this. Thus, our list of requirements is intentionally short.

Feedback Agent Requirements

1. **Provide the current mobility metric value for the mobile node.**

   When queried, the feedback agent must provide a value for the current mobility experienced by the node. Our implementation provides the value for the node's average link duration in seconds.

2. **Provide selection of active or passive monitoring.**

   There is an obvious tradeoff between the accuracy of the feedback mechanism and the resources expended to gather the information. Passively monitoring link durations requires little or no additional resources, but may result in widely differing accuracy. Active monitoring can provide highly accurate duration information at the cost of increased resource consumption. The protocol(s) using the feedback agent should be able to specify the monitoring technique. Section 5.2 provides more detailed discussion of this tradeoff as well as implementation decisions.

3. **Allow real time parameter customization.**

   The gathering of link duration information by the feedback agent is influenced

by three key parameters. The feedback agent allows protocol(s) to adjust these parameters to achieve the desired balance between accuracy and overhead. These parameters and their implemention specifics are discussed in Section 5.4.

5.2 Passive vs. Active Monitoring

A node can gather link durations in a resource efficient, distributed manner by passively recording received packet information. In other words, every time a mobile node hears a transmission from a neighboring node, the node records the link availability. Over time, communicating with or over hearing new found neighbors signals link additions, and time stamps coupled with an aging mechanism signal link breakages and therefore link durations. Passively gathering link duration information has the obvious benefit of adding only a negligible processing burden and minimal state information to each node with **no** increase in network overhead.

The key to successful feedback from passive monitoring is that if network traffic is high, and many or all nodes are involved in network communication, then the performance and accuracy of passive monitoring approaches the actual link durations of the network. This is precisely the demanding network condition (high congestion and contention) that needs the most feedback and protocol adaptation. In times of light traffic between relatively few network nodes, passively gathered link duration information is less accurate; however, an ad hoc network protocol is minimally stressed

by this environment and would probably benefit little from adaptation.

Active monitoring of link durations can be done either with link layer support (hello beacons) or directly as an adaptive feedback mechanism. Active monitoring has the obvious cost of increased communication, which could be unacceptable to nodes in terms of resource usage (e.g., battery, etc.). However, active monitoring results in much more accurate link duration information, and thus the ability to adapt more effectively.

A possible middle ground is the combination of active and passive monitoring of link duration information. In times of low network traffic an active monitoring technique can be effective, especially if network mobility is high. Any time network traffic is high, regardless of network mobility, passive monitoring can be effective. Our implementation of the feedback agent includes both passive and active monitoring. In addition, it is possible to switch between the two manually or automatically based on the current network environment.

When passive monitoring is selected the feedback agent monitors every packet received by the link layer and records the reception time and sending node's ID. No beaconing takes place. When active monitoring is selected, the feedback agent periodically broadcasts a network layer beacon which includes the sending node's unique ID. The byte overhead associated with this beaconing is minimal since the beacons are very small broadcast packets which contain only the sending node's ID.

To minimize the feedback agent's packet overhead due to beaconing, promiscuous receptions take the place of beacons even if the feedback agent is operating in active mode. Specifically, the agent resets the beacon timer on a node every time any packet is transmitted. This packet transmission is assumed to take the place of the beacon. Therefore, the higher the network traffic, the lower the overhead associated with our active feedback beaconing.

5.3 Feedback Agent Operation

When passive mode is selected and the beacon timer expires, no action is taken[1]. When active mode is selected and the beacon timer expires, a network layer feedback beacon is broadcast. This packet contains only the sending node's unique ID. The feedback agent operates in the same manner at the reception of every packet: when a packet is received, the sending node's unique ID and the reception time are recorded.

When queried by another protocol on the mobile node, the feedback agent can provide the following information:

- Current link duration: an average duration of all the links which have *ended* within the last duration window. Active links are not included in this calculation.

- Total average link duration: the average duration of all the node's links which

[1]In passive mode, the beacon timer is used to facilitate the calculation of a node's current link duration.

have ended.

- Current number of neighbors: the current number of neighbors the node has.
- Total average number of neighbors: the average number of neighbors the node has had for the lifetime of its operation.
- Current neighbors: a list of the node's current neighbors and the length of time their links have been *active*.

5.4 Key Parameters

The calculation of the information provided by the feedback agent depends on three key parameters:

- beacon period,
- link break time, and
- duration window size.

These parameters and implementation details, which provide more insight into the operation of the feedback agent, are discussed in Sections 5.4.1 - 5.4.3.

5.4.1 Beacon Period

This parameter controls the frequency of the feedback beacon during active mode operation. There is obviously a trade off between overhead and accuracy. Initial

testing revealed that the beacon period needs to be much smaller than the actual duration being measured.

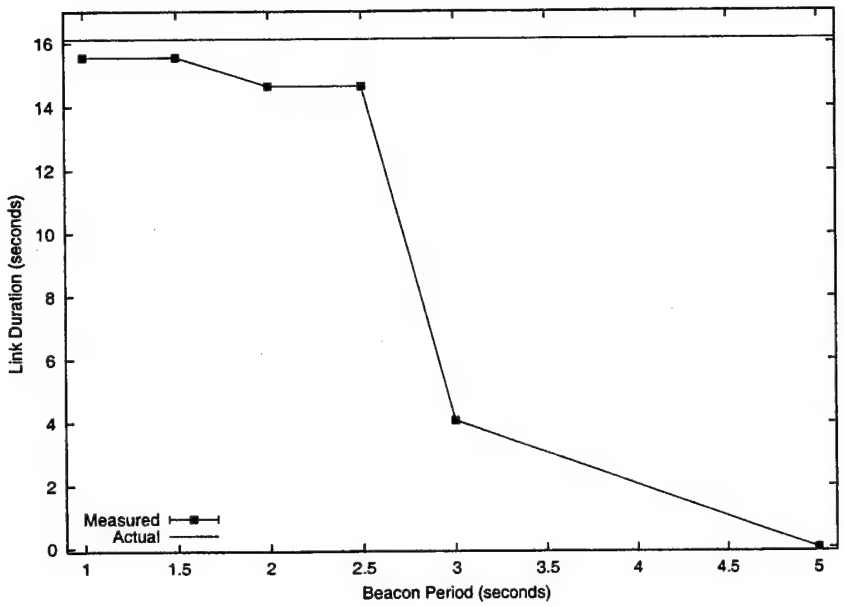

FIG. 5.1. Feedback Agent Average Measured Link Duration vs. Beacon Period.
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Link break time = 3 seconds, duration window size = 30 seconds.

To investigate the effect of beacon period on link duration accuracy, we chose a mobility scenario with a link duration of 16.125 seconds. This link duration corresponds to a random way-point mobility scenario with node speeds of 10 m/s and pause times of 10 seconds. Figure 5.1 shows the effect of the measured link duration with various beacon periods; as the beacon period is lengthened the measured duration is

less accurate. We note that the congestion and contention created by beacons under one second make simulation infeasible. The sudden decrease in measured accuracy at the three second beacon period is a result of the link break time parameter. This parameter and the effect are described in Section 5.4.2.

Our emphasis is on improving a protocol's operation in very challenging mobility scenarios. These high mobility scenarios are characterized by link durations of only a few seconds. The beacon period must be suitably short in order to provide accuracy in these instances. Thus, we set the beacon period at *one second* in the rest of our simulations.

5.4.2 Link Break Time

The link break time parameter indicates how much time can pass without traffic on a current link before that link is declared broken. If this value is too large a link may break and then reconnect without being detected, which leads to artificially long link durations. On the other hand, if this value is too small, dropped packets due to contention appear as link breakages, which leads to artificially short link durations. The link break time parameter must be shorter than the shortest anticipated link duration. In our highly mobile scenarios link durations of only a few seconds are possible.

Figure 5.1 shows a dramatic drop in measured link duration accuracy at three seconds, which occurs because the link break time used was three seconds. The

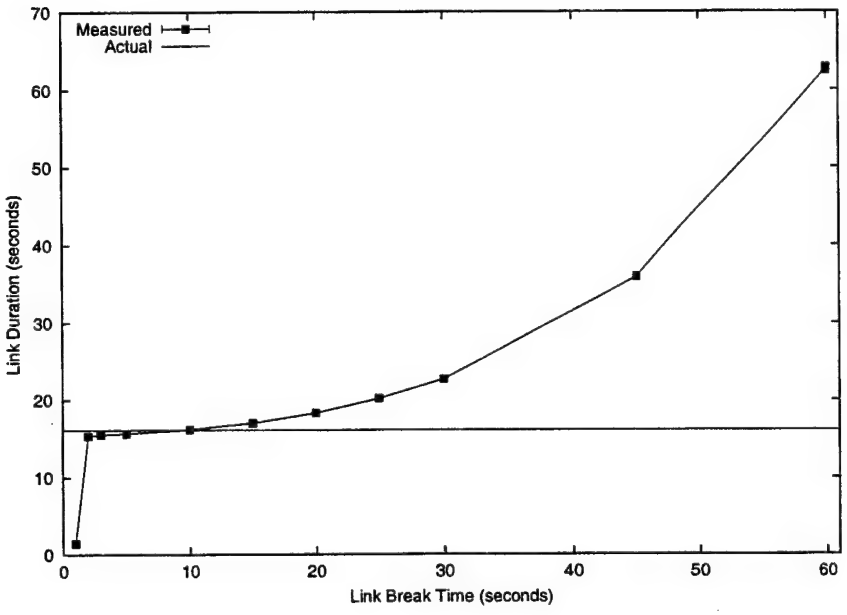

FIG. 5.2. Feedback Agent Average Measured Link Duration vs. Link Break Time.
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Beacon period = 1 second, duration window size = 30 seconds.

performance drop signifies the inter-dependence of beacon period and link break time.

Using a one second beacon period (see Section 5.4.1), Figure 5.2 shows the result on measured link duration for various values of the link break time. The value of link break time must be longer than the beacon period to dampen the affect of a few missed beacons, and shorter than the shortest anticipated link duration (3.5 seconds in Table 4.1). In light of these competing requirements, and the results of Figure 5.2, we choose *three seconds* as the value of our link break time parameter. This value

allows for the measurement of link durations as short as three seconds, while allowing up to two beacons in a row to be missed.

5.4.3 Duration Window Size

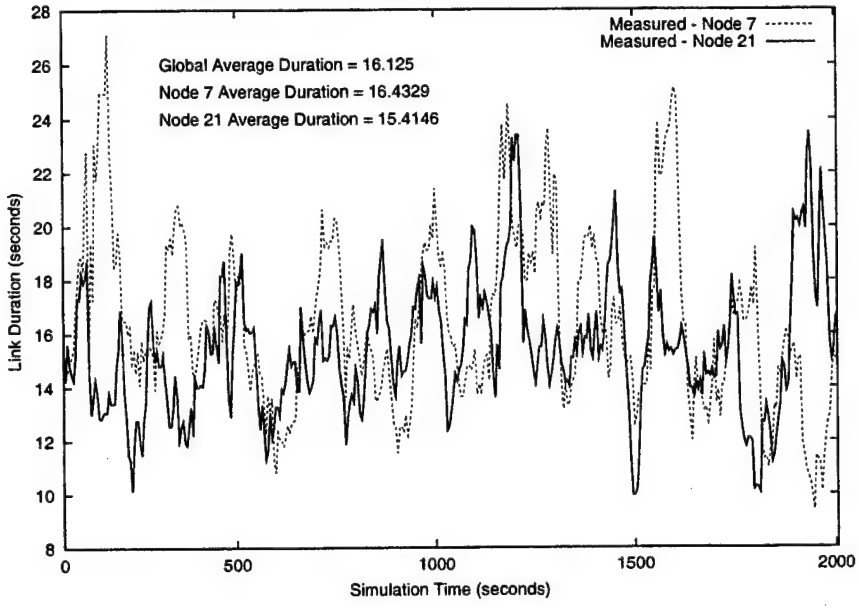

FIG. 5.3. Feedback Agent Current Measured Link Duration vs. Simulation Time (Nodes 7 and 21).
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

The duration Window Size (WS) parameter is the length of time the feedback agent uses to average the previously heard links when determining the current link

duration. Figure 5.3 shows the link duration of two mobile nodes sampled at five second intervals for a 2000 second simulation scenario. Three observations are apparent. First, there is a large variability in a node's link duration. Both nodes in the plot vary above and below their mean duration by at least one third. Second, there is a significant difference in the link duration experienced by different nodes at the same time. And third, a node's link duration can change very rapidly.

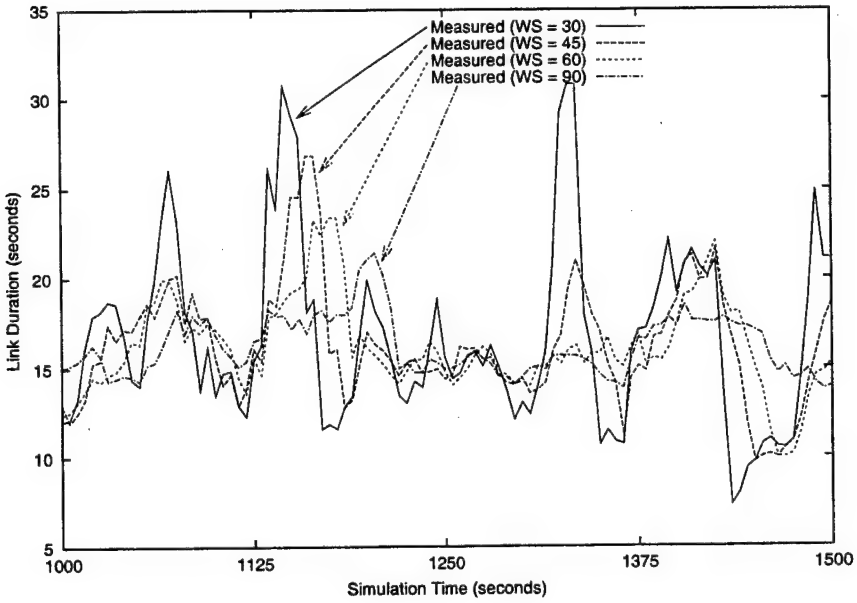

FIG. 5.4. Feedback Agent Current Measured Link Duration Using Differing Window Sizes (Node 21).
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Beacon period = 1 second, link break time = 3 seconds.

Figure 5.4 illustrates actual and measured link duration for node 21 with four

different window sizes over a period of 500 seconds of simulation time. We note the peaks of the four curves indicated by the four arrows in the figure. All four curves represent the same peak in link duration. The feedback agent using the smallest duration window size (30 seconds) reports the highest peak first. In other words, the smallest window size is the most responsive and shows the most variability. The feedback agent using the largest duration window size (90 seconds) reports the lowest peak last. In other words, the largest window size is the least responsive and shows the least variability.

Investigating smaller values for the duration window size indicates that the feedback agent becomes overly sensitive and reports extreme variation (compare Figures 5.4 and 5.5). We note the increased range on the y-axis in Figure 5.5 (0-45 seconds vs. 5-35 seconds). The value chosen for duration window size embodies another tradeoff. If the value is too large the feedback will be unresponsive to changes in the current link duration. If the value is too small the feedback will be overly sensitive to variations in the link duration. We compromise by setting the window size at *30 seconds* in the rest of our simulations.

5.5 Feedback Agent Accuracy

We assess the accuracy of the feedback agent with our chosen parameters in Figure 5.6. This figure illustrates the measured and the actual link durations for four

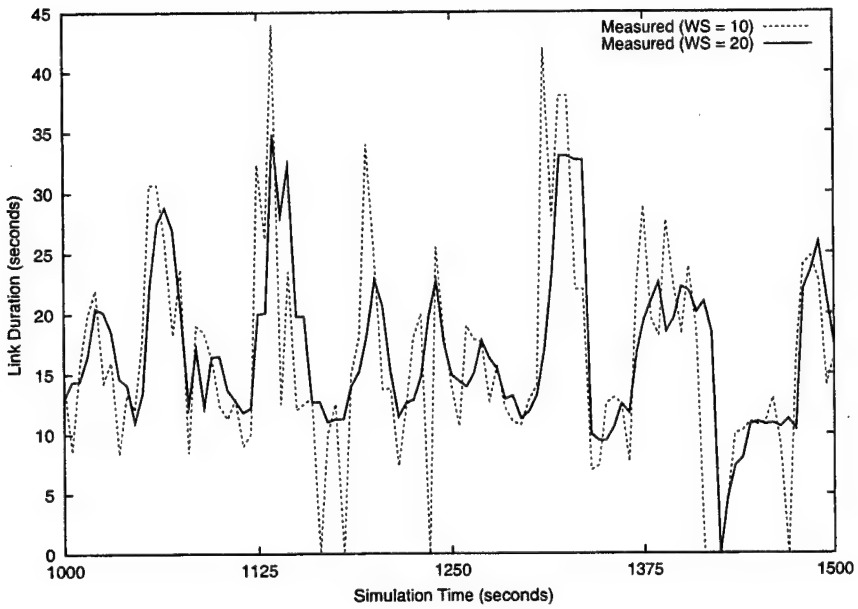

FIG. 5.5. Feedback Agent Current Measured Link Duration for Smaller Window Sizes.
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Beacon period = 1 second, link break time = 3 seconds.

movement scenarios and ten simulation trials. The feedback agent consistently reports the link duration as lower than the actual link duration. We find this pessimistic estimation of the feedback agent to be advantageous. The purpose of the feedback is to signal when it is appropriate for protocols to adapt to mobility. Consistent reporting of the link duration as shorter, or the mobility and networking dynamics as more demanding, is preferred to under estimating the actual demands of mobility.

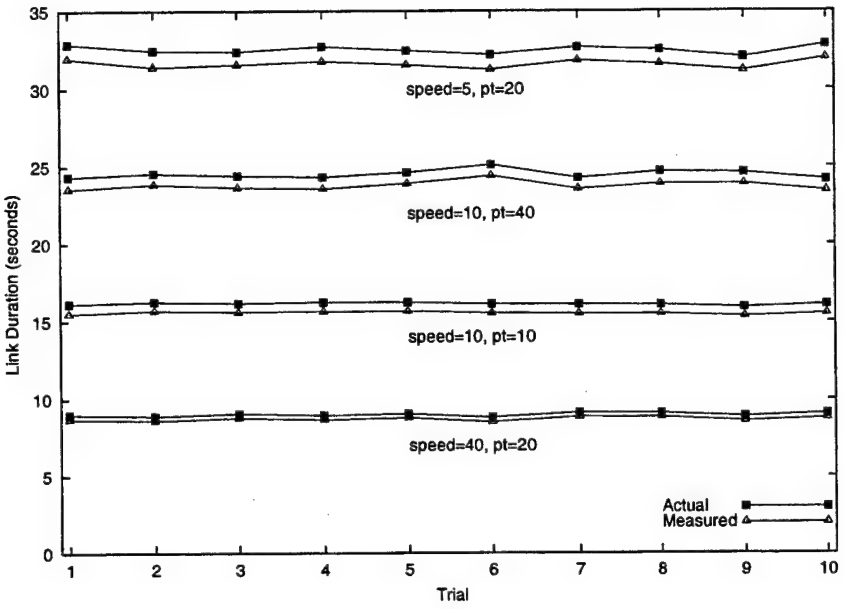

FIG. 5.6. Feedback Agent Average Measured Link Duration for Four Simulation Scenarios vs. Ten Trials.
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

5.6 Conclusions

The data presented in [5] and Chapter 3 are a result of analyzing network scenarios with global information. As a result, the link durations reported are actual link durations experienced by each node and the values in the figures are a global average of the link duration experienced by all nodes. The results presented in this chapter illustrate the feasibility of accurately gathering link duration information in a totally

distributed manner. Specifically, it is possible for an individual node to accurately monitor the link duration it is experiencing.

Chapter 6

ADAPTIVE LOCATION AIDED ROUTING FROM MINES (ALARM)

We use our link duration feedback agent (see Chapter 5) to create an adaptive MANET unicast routing protocol which we discuss in this chapter. The protocol chosen for adaptation is the Location Aided Routing (LAR) protocol originally described in [36] and further refined, tested, and evaluated in [11] (see Section 2.1.2). We choose to apply feedback and enable adaptation of an existing protocol to demonstrate the effectiveness of adapting based on meaningful feedback. The goal is not to create Yet Another MANET Protocol (YAMP). Instead, the goal is to optimize existing protocols and to create a mechanism to enable the combination of multiple protocols into a hybrid protocol. Our hybrid protocol uses the most effective protocol technique based on the network conditions currently being experienced by a given mobile node. Ironically, for simplicity, we refer to this hybrid protocol development as a MANET protocol.

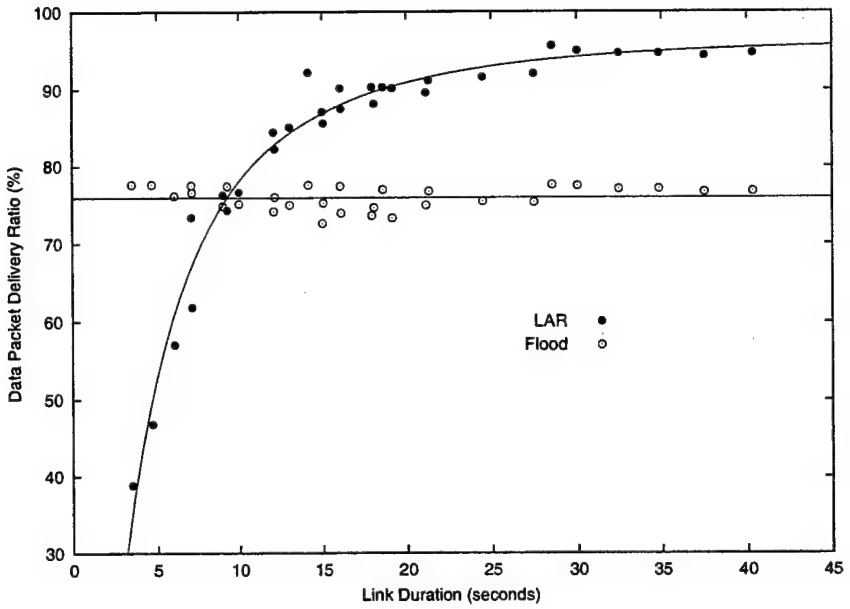

FIG. 6.1. Data Packet Delivery Ratio - LAR and Flood
Speed = [5,10,20,30,40] m/s, pause time = [0,10,20,30,40,50] seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.

6.1 Approach

Our initial approach to protocol adaptation is to combine the LAR protocol with a directed flooding method[1]. Figure 6.1 indicates that LAR is effective at delivering packets if the link duration is longer than 10 seconds. When network dynamics create links that break quickly, LAR is unable to "keep up with" topology changes.

The most simple fall-back method for a MANET routing protocol is flooding.

[1]Throughout this discussion, we use the terms directed flooding and box flooding interchangeably.

In the case when location information is available to the protocol, directed flooding can be used. Directed flooding consists of flooding the data packet in a box oriented in the direction of the destination. Such a box can be determined based on the last known location of the destination, in a manner similar to directed flooding of route request packets in LAR (see Section 2.1.2).

6.2 ALARM Overview

Our combined or hybrid protocol, Adaptive Location Aided Routing from Mines (ALARM), uses link duration feedback at each node to determine the appropriate forwarding method for data packets. When link durations of nodes on the source route in the packet header are longer than a predetermined threshold, ALARM forwards data packets along this source route (i.e., ALARM uses LAR). When link durations of a node on the source route are shorter than the threshold, that node initiates a directed flood of the data packet toward the destination. This flood is automatically "dampened" when the flooded packets reach nodes that have link durations longer than the threshold.

There are three possibilities for a packet which was flooded on its last hop. First, if the packet reaches a node which has link durations below the threshold (signaling an unstable area of the network), the packet continues to be flooded. Next, if the packet reaches a node which has link durations longer than the threshold (which signals a

more stable network), one of two options is possible. If the node is on the original source route, the packet continues on the source route toward the destination. On the other hand, if the packet reaches a node with long link durations which was not on the original source route, the packet is dropped. The emphasis is on delivering packets through regions of high network instability. Either the packet reaches the final destination via directed flooding, or it picks up the original source route after flooding through a network "hot spot".

One additional detail included in ALARM is a flood horizon. The flood horizon signifies how many hops a packet continues to be flooded past the mobility hot spot. This flood horizon increases the chance that either the destination or a node on the original source route is reached. Obviously, a tradeoff between increased overhead and increased reliability exists.

The node that initiates the directed flood (i.e., the first node reached with short link durations) unicasts an ALARM packet, via the reverse source route, to the data packet source. It is then possible for the source node to take preemptive action, such as initiating a new route discovery. Initiating a new route request before a route error is received was first discussed in [20] for DSR. In [20] low signal strength (not link duration between two nodes on the source route) is the trigger for a preemptive action.

In reactive protocols such as DSR and LAR, routes must be repaired when a link

on an existing route fails. Route failures are signaled by a route error mechanism. When this occurs, the current packet that experienced the route error is usually lost. ALARM allows for the successful delivery of these packets by flooding the packets through network mobility "hot spots". We have shown that delivery performance can be dramatically improved in highly dynamic networks by "saving" the packets that are normally dropped when a link on a source route breaks (see Conclusion 5 in Section 2.5).

6.3 ALARM Parameters

The following parameters are required for the ALARM protocol to operate. See Chapter 5 for details on the first two parameters.

- Duration Window Size - the moving window, in seconds, that the link duration is averaged by the feedback agent,
- Link Duration - current link duration averaged over the last duration window seconds,
- ALARM Threshold - when the link duration falls below a threshold, the protocol adapts, and
- Flood Horizon - the number of hops that a flooded data packet continues to flood past a mobility "hot spot".

The duration window size is supplied to the feedback agent which in turn reports back the current link duration for the node. The link duration supplied is a measure of the average link duration for all the node's one hop neighbor links. Based on the feedback agent discussion in Section 5.4 we use a duration window size of 30 seconds. ALARM threshold and flood horizon are the key ALARM parameters. A detailed discussion of protocol optimization using these parameters occurs in Section 6.5.

6.4 ALARM Details

This section presents the ALARM protocol algorithm as pseudo code. We discuss both the protocol and implementation details.

Sender

Send Data Packet

```
if (LinkDuration < ALARMThreshold)
{
  if (no source route)
  {
    Initialize source route to Null
  } else {
    Use source route from local route table
  }
  Initialize FloodHorizon
  Box flood data packet
} else {
  Initialize FloodHorizon to zero
  use LAR algorithm to send packet
}
```

Receive ALARM packet

```
Increment number of received alarms for node which sent alarm
if (alarms[node] > ALARM route error trigger)
{
  send Route Request packet
}
```

We note the following features in the sending algorithm:

- The sending node (source of a data packet) determines if it is experiencing high network mobility (short link durations). If it is, then it uses a box flood to send the data packet. Otherwise, the packet is sent by LAR.

- If there is an existing *good source route*, one which is recent and has not experienced a previous route error, ALARM includes the route within a flooded data packet.

- If the sending node does not have a *good source route*, then the data packet is box flooded with a null source route. The LAR route request/reply mechanism has little chance of success when a sending node is experiencing very short link durations.

We note the following action by the sending node upon receipt of an ALARM packet:

- When a sending node receives an ALARM packet, it records which node sent the packet.

- When some threshold of ALARM packets have been received, an LAR-style route request is triggered. We use a value of *three* received ALARMs from one intermediate node as an indication of an imminent route break. This condition invalidates all the routes which use that link and triggers a route request/reply cycle for the route and destination affected. When a newer route is discovered for a destination, for instance by promiscuous listening, the ALARM counter is reset for each node on the new route.

Forwarding Node

Receive Packet for Forwarding

```
if (PacketType in [routeRequest, routeReply, routeError, ALARMpacket])
{
  use LAR algorithm to forward or reply to packet appropriately
} else if (PacketType == Data) {
  if (source route is Null)
  {
    if (source route in local route table)
    {
      Add route to data packet header
    } else {
      Initialize FloodHorizon
      Box flood data packet
      return
    }
  }
  if ((LinkDuration < ALARMThreshold) or (routeError detected))
  {
    if (this is the first node to initiate box flood)
    {
      Send ALARM packet to source of data packet
    } else if (routeError detected) {
      Send routeError packet
    }
    Initialize FloodHorizon
    Box flood data packet
  } else {
    if (FloodHorizon > 0)
    {
      Decrement FloodHorizon
      Box flood data packet
    } else {
      use LAR algorithm to send packet
    }
  }
}
```

We note the following features in the forwarding algorithm:

- If the packet is not a data packet, handle it using LAR.
- If the source route in the received data packet is null, check the local route table for a route. If a *good* route is in the local route table, insert it into the data header and continue with the ALARM algorithm. If there is not a *good* route to the destination in the local route table, re-initialize FloodHorizon and continue box flooding the data packet.
- If the data packet is received by a node that is experiencing network instability or has detected a route error, re-initialize the FloodHorizon and box flood the data packet. If this node is the initiator of the box flood, send an ALARM packet to the data packet originator. If this node detected the route error, send a route error packet to the data packet originator. We note that every node experiencing network instability re-initializes FloodHorizon. Thus, the dampening procedure begins when the packet has passed the mobility "hot spot".
- If the flood horizon is greater than zero, the node is not experiencing network instability, and has not detected a route error, continue to box flood the packet and decrement the flood horizon.
- If a data packet is received by a node that is not experiencing network instability,

the packet's flood horizon equals zero, and the node is not on the original source route, drop the packet. In other words, the flood is dampened.

- If a data packet is received by a node that is not experiencing network instability, the packet's flood horizon equals zero, and the node is on the original source route, the data packet is unicast to the next node in the route via LAR.

6.5 Protocol Parameter Optimization

The primary parameters in the ALARM protocol are the ALARM threshold and flood horizon. The other essential parameters (duration window size and link duration) are obtained directly from the feedback agent (see Section 5.4). The following three points provide challenges in determining appropriate values for the two ALARM parameters. First the two parameters are inter-dependent. They are also sensitive to the actual link duration, data load, data traffic distribution, etc. Finally, since we want to optimize ALARM for all different protocol options, the process of determining values for protocol parameters becomes even more complex.

Another difficulty with a complex protocol is selecting which performance metrics to maximize, and what is considered an acceptable cost for the improvement of each performance metric. Requirement 2 in Section 3.4 lists three performance areas:

1. data packet delivery ratio,

2. end-to-end delay, and

3. protocol overhead.

The consideration of protocol overhead is further divided (see Section 3.4) and requires evaluation of packet and byte overheads for control, data, and total transmissions. While our primary focus is on data packet delivery ratio, we discuss the cost (i.e., protocol overhead) in detail as well. Results for delay are presented; however, it is the least affected performance result.

We optimize ALARM using a moderate link duration of 16.125 seconds, which corresponds to a random way-point mobility model scenario of 10 m/s node speed and 10 second pause times. We evaluate five different values for the ALARM threshold, $[0, 3, 6, 9, 12]$ seconds, and five different values for the flood horizon, $[0, 1, 2, 3, 10]$ hops. Finally, there are three ALARM protocol options which we evaluate. These are described below with the corresponding label used in the following figures and discussion.

1. ALARM-all: the ALARM mechanism is used by all nodes, both sending and forwarding.

2. ALARM-fwd: the ALARM mechanism is enabled only on forwarding nodes and not the sending node.

3. ALARM-err: the ALARM mechanism is disabled except when a route error occurs.

The above combination of possibilities creates a "performance volume" for the ALARM parameters (see Figure 6.2). This volume has the ALARM threshold on the x-axis, the flood horizon on the y-axis, and the performance metric of concern on the z-axis. Each volume includes three surfaces which correspond to the protocol options enabled. We note that on all figures each data point represents an average of ten simulation trials. A 95% confidence interval was calculated for each point, and in all cases the intervals are quite small. These confidence intervals are not shown on the figures in order to increase their clarity.

6.5.1 Data Packet Delivery Ratio

Figure 6.2 shows the data packet delivery ratio for the possible values of our ALARM parameters and protocol options. The figure shows a decrease in performance for the ALARM-fwd and ALARM-all protocol options as the ALARM threshold increases. It also shows a clear ranking of the protocol options with ALARM-err performing the best, ALARM-all the worst, and ALARM-fwd in between.

We note that the only versions of ALARM-fwd and ALARM-all that perform well, compared to ALARM-err, are when the ALARM threshold is set to zero seconds (see Figure 6.2). Examining the algorithm in Section 6.4 reveals that both ALARM-fwd and ALARM-all operate essentially the same as ALARM-err when ALARM threshold is zero, i.e.,

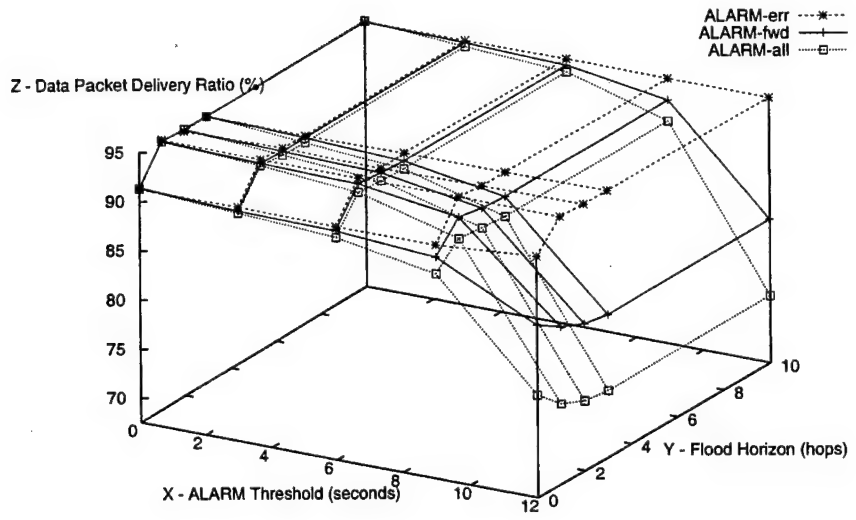

FIG. 6.2. ALARM Performance Volume - Data Packet Delivery Ratio
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

```
(LinkDuration < ALARMThreshold)
```

can never occur. In other words, ALARM-fwd and ALARM-all only initiate a box flood of a data packet when a link error is detected.

Figure 6.3 extracts data from Figure 6.2 in order to aid us in understanding the ALARM threshold. ALARM-fwd and ALARM-all (with flood horizon equal to one and ten) both perform worse as the ALARM threshold increases while ALARM-err

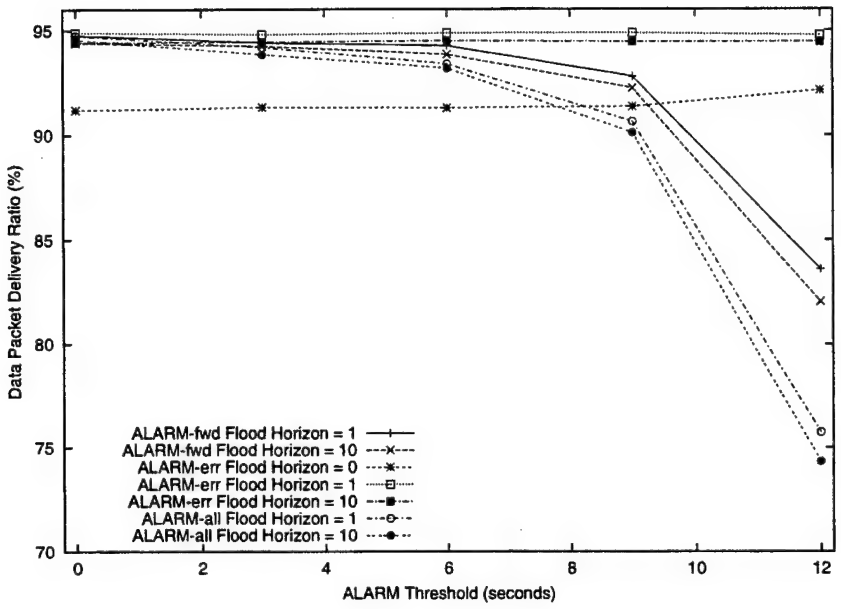

FIG. 6.3. ALARM Data Packet Delivery Ratio vs. ALARM Threshold
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

(regardless of the flood horizon) is not affected by the ALARM threshold value. Both ALARM-fwd and ALARM-all base a significant protocol decision on the ALARM threshold value, that is, whether to initiate a box flood or not. ALARM-err, on the other hand, makes flooding and dampening decisions based on link breakage only, not on the mobility value (link duration) experienced. As the ALARM threshold increases, ALARM-fwd and ALARM-all behave as a pure flooding protocol. Thus,

their performances decrease appropriately.

Figure 6.3 also illustrates the effect of the flood horizon on the best performing protocol option, ALARM-err. A value of zero for flood horizon, which means a data packet encountering a link error is box flooded for only one hop, performs worse than when the flood horizon is increased. The other two values of flood horizon shown (one and ten) have similar delivery percentages. Other non-zero values of flood horizon perform similarly as well. In summary, ALARM-err performance improves when flood horizon is greater than zero. In the next section we examine the effect of flood horizon and ALARM threshold on protocol overhead.

6.5.2 Overhead

Figure 6.4 shows the protocol overhead associated with the various versions of the ALARM protocol[2]. As before, ALARM-err is unaffected by the ALARM threshold, while the protocol overhead for ALARM-fwd and ALARM-all increases as the ALARM threshold increases.

Figure 6.5 examines the protocol overhead of ALARM-err in more detail. This figure reveals that there is a reduction in overhead per data packet delivered when a box flooded data packet is allowed to travel past the network "hot spot" before the flooding is dampened, i.e., for values of the flood horizon greater than zero. As

[2]While we focus on control, or protocol packet overhead, all other overhead types (total packet and byte overhead) follow the same patterns. In other words, the observations and conclusions made by examining the control packet overhead are equally applicable to the other forms of overhead.

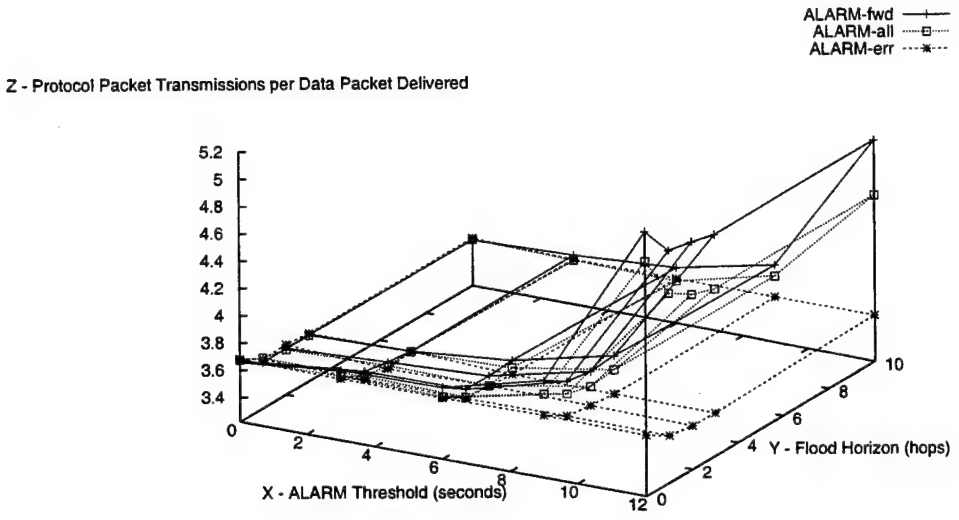

FIG. 6.4. ALARM Performance Volume - Protocol Packet Transmissions per Data Packet Delivered
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

seen in Section 6.5.1, using zero hops for the value of flood horizon results in fewer delivered data packets, so the overhead ratio is higher.

While protocol packet overhead is comparable for all values of the flood horizon greater than zero (see Figure 6.5), Figure 6.6 demonstrates that greater values of the flood horizon increases the number of packets in the system due to data packet flooding. This extra flooding has been shown to have little or no benefit to delivery

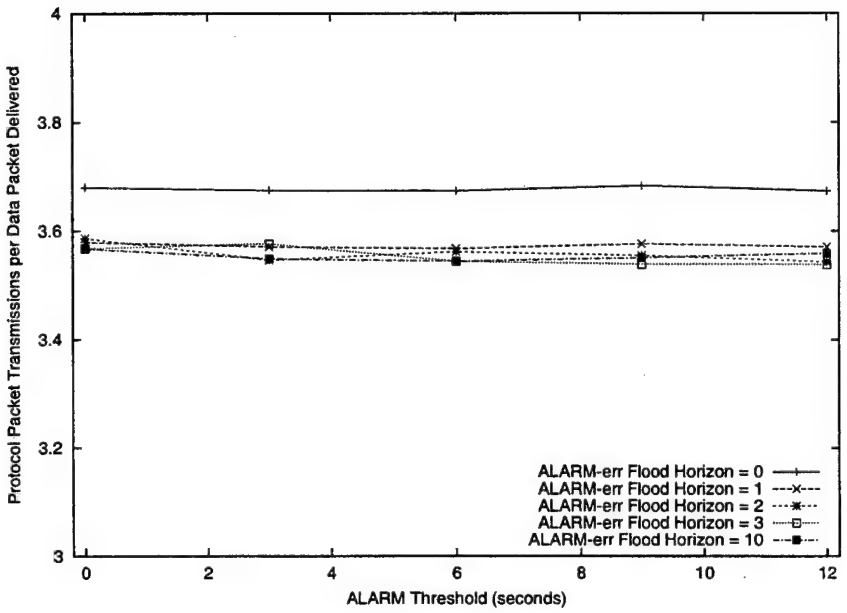

FIG. 6.5. ALARM Protocol Packet Transmissions per Data Packet Delivered vs. ALARM Threshold
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

ratio (see Section 6.5.1 and Figure 6.3).

6.5.3 End-to-End Delay

Figure 6.7 presents the end-to-end delay of the ALARM variants. Examining the figure reveals that end-to-end delay is stable for all three protocol options as both the flood horizon and ALARM threshold change. In all cases, end-to-end delay

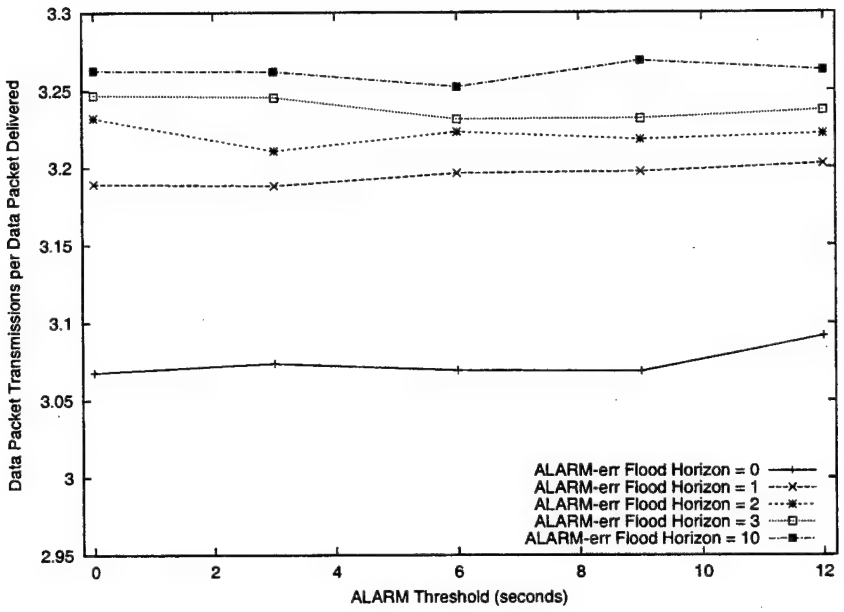

FIG. 6.6. ALARM Data Packet Transmissions per Data Packet Delivered vs. ALARM Threshold
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

only varies between 0.1680 and 0.2766 seconds. Furthermore, the range of delay for ALARM-err, the variant of choice, is only 0.2295 to 0.2759 seconds.

In conclusion, ALARM-err demonstrates the best data packet delivery ratio, the lowest overhead, and acceptable delay. Thus, we use ALARM-err in our comparison to LAR and Flood. Our results indicate to set ALARM threshold to *nine seconds* and flood horizon to *one hop*. These values are an effective combination to ensure

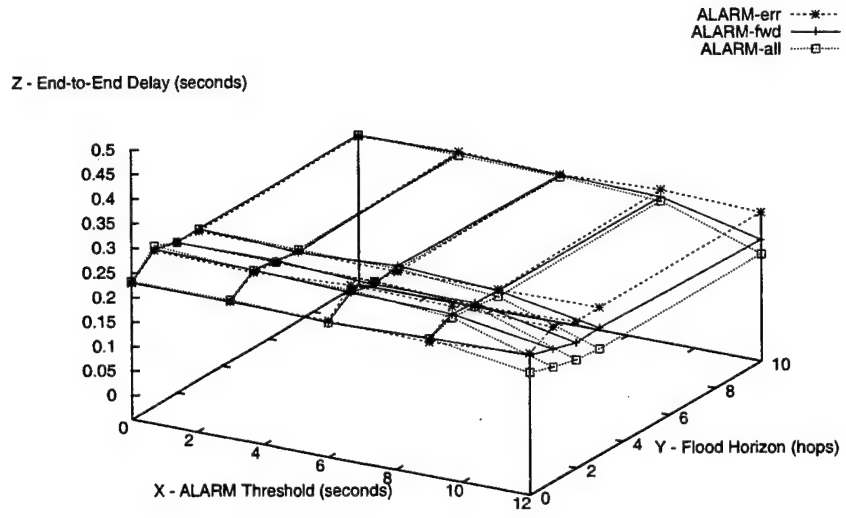

FIG. 6.7. ALARM Performance Volume - End-to-End Delay
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.

high data packet delivery and low overhead.

6.6 Performance Comparison: ALARM, LAR, and Flood

Our initial goal was to combine the strengths of the LAR protocol in mild to moderate mobility with the ability of flooding to effectively deliver data packets in high mobility (see Figure 6.1). Figure 6.8 demonstrates we have met our goal. The

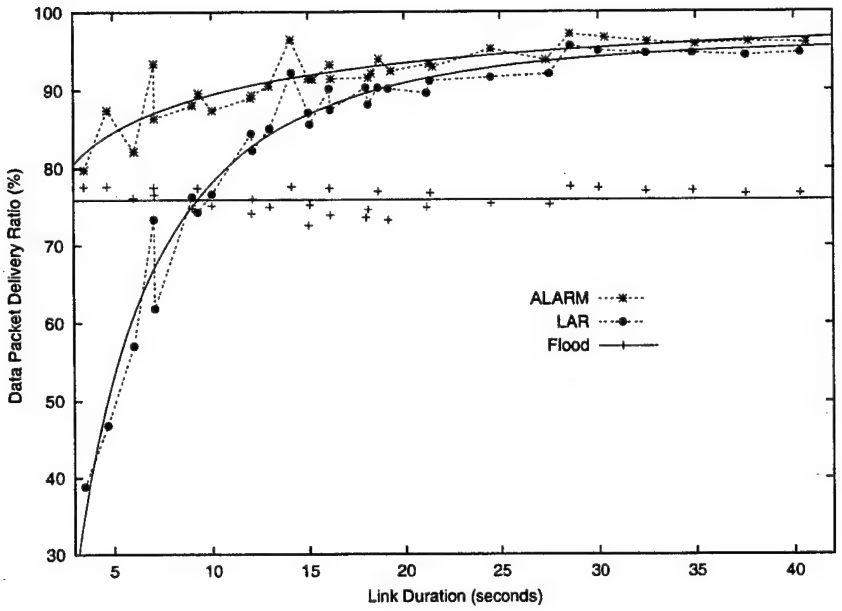

FIG. 6.8. Data Packet Delivery Ratio - ALARM, LAR, and Flood
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

combination of directed flooding with Location Aided Routing (LAR) creates a hybrid protocol which can out-perform either component protocol.

The key to our success is "saving" the packets normally lost due to route errors. Specifically, we initiate a box flood toward the destination when a route error occurs. Figure 6.8 demonstrates that ALARM improves the delivery ratio, and Figure 6.9 shows that ALARM reduces protocol overhead, compared to both LAR and Flood.

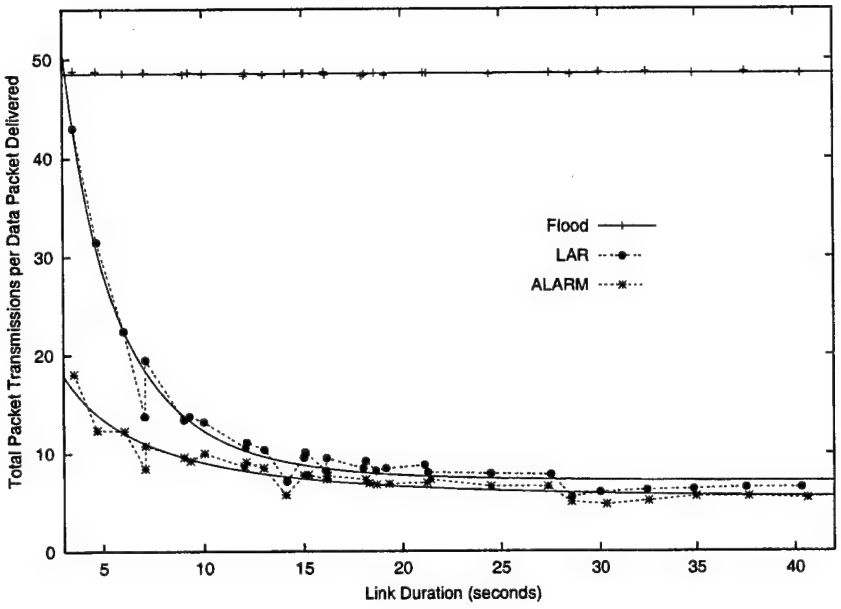

FIG. 6.9. Total Packet Transmissions per Data Packet Delivered - ALARM, LAR, and Flood
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

The raw overhead of ALARM is similar to LAR; however, since ALARM delivers significantly more packets, the protocol overhead per data packet delivered is smaller for ALARM. We note that Figure 6.9 reports total packet transmissions since Flooding has no control packets. The entire cost associated with Flooding is only due to duplicate data packet transmissions.

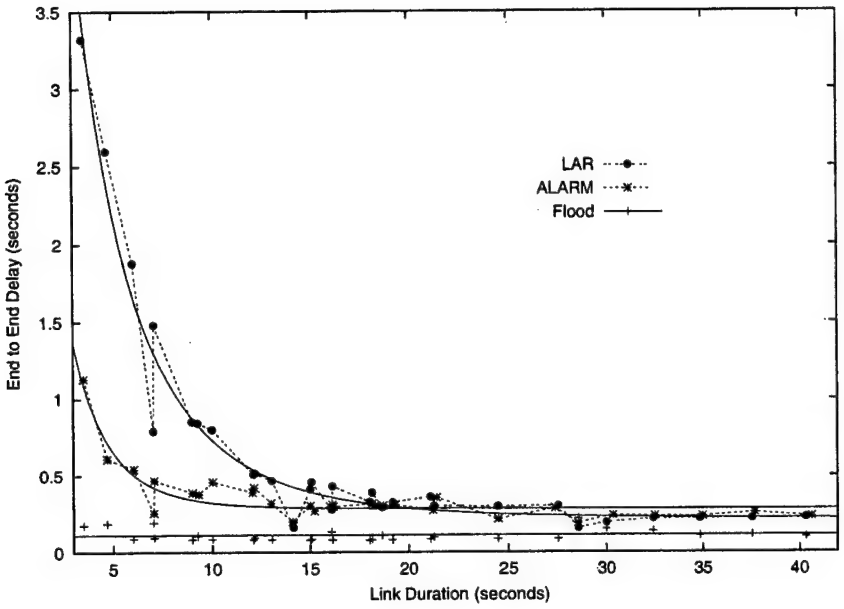

FIG. 6.10. End-to-End Delay - ALARM, LAR, and Flood
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

Figure 6.10 shows the end-to-end delay for the ALARM, LAR, and Flood protocols. Flood has the lowest delay since no route request/reply cycle exists. ALARM improves upon the delay of LAR, especially when mobility is high (link durations are short). When mobility is low, ALARM and LAR have similar delays and are both comparable to the delays provided by Flood.

6.7 Conclusions

The development of a hybrid protocol by combining existing protocols is an effective way to optimize protocol performance over a wide range of network scenarios. We developed an effective hybrid protocol (ALARM) which is superior to both component protocols in data packet delivery and overhead. It combines the low overhead of LAR in times of mild to moderate mobility with the high delivery ratio of Flood in times of high mobility. In summary, ALARM has lower overhead and higher delivery ratios than LAR and Flood for a wide range of mobility scenarios.

We note the following conclusions. First, combining two protocols (in our case, one simple and one complex) results in a much more complex protocol. Not necessarily more complex to implement or operate, but much more complicated to optimize and understand.

Next, recall that in ALARM-err, nodes only begin to box flood data packets when a route error occurs. The other two ALARM options preemptively box flood data packets when short link durations are experienced, regardless of whether or not the source route link is intact. Figure 6.2 reveals that this preemptive flooding is always a mistake. If a source route link is still active, a protocol should always use it. The full benefit that can be derived from mobility (link duration) feedback in proactive and hybrid protocols remains an open question.

Finally, the feedback agent of Chapter 5 is inherently a proactive networking

element. Coupling this proactive element with a completely reactive protocol such as LAR was not as beneficial as expected. Specifically, the performance improvements seen by ALARM are due primarily to an optimization of LAR which is independent of the feedback agent (i.e., saving packets normally discarded at intermediate nodes due to link failures). However, our feedback agent proved beneficial in dampening of the directed flood. Our dampening mechanism allows us to obtain higher deliver ratios with minimal cost.

Chapter 7

MATHEMATICAL MODEL FOR RELIABILITY, OVERHEAD, AND DELAY

This chapter presents our development of mathematical models based on link duration. The general approach is similar to that taken in [29] and [30]. The key difference is the use of link breakage as the mobility metric in [29] and [30], and the use of link duration as the mobility metric herein. In addition, the focus of [29] and [30] is only on protocol overhead. We develop models for data packet delivery ratio, overhead, and delay using link duration as the primary metric.

We consider link duration as the average Mean Time To Failure of the wireless link (T_{link}). Suppose L represents the average route length, and T_{link} represents the average link duration. We define T_{route} as the Mean Time To Failure of the entire route; thus T_{route} can be calculated as $\frac{T_{link}}{L}$[1]. For example, a common T_{link} for moderate mobility is approximately 60 seconds. If the average route length is five hops, the resulting T_{route} is 12 seconds. Table 7.1 defines the variables for our performance models. In Sections 7.1 through 7.5 we define our models. We then compare the simulated values of the protocols and the expected values from our

[1] $T_{route} = \frac{T_{link}}{L}$ is true if the assumption can be made that all link failures are exponentially distributed and occur as independent events. This simplifying assumption is made for the results presented in this chapter.

T_{link}	average link duration
L	average route length
T_{route}	average route duration
N	number of nodes in the network
M	number of edges (wireless links) in the network
$\Delta = \frac{2M}{N}$	average node degree (number of neighbors)
C	number of data flows in the network
α	packet arrival rate per flow per second
R	data packet delivery ratio (Reliability)
R_{LAR}	data packet delivery ratio of LAR
R_{ALARM}	data packet delivery ratio of ALARM
P_{lost}	the number of packets lost when a route error is received by the source
T_{sim}	simulation time
T_{tx}	unicast packet transit delay per hop
D_u	unicast packet delay per hop
D_f	flooded packet delay per hop
F_{raw}	flooded packet delivery ratio, no other network traffic present
F_r	flooded packet delivery ratio, other protocol traffic present
P_f	probability of channel access failure
O_{total}	total (data and control) protocol packet overhead
D_{base}	base end-to-end delay without pending packet queue waiting time
$Q_{average}$	average length of the pending packet queue (in seconds)
Q_{wait}	time data packets wait in the pending packet queue
D_{LAR}	end-to-end delay experienced by delivered LAR packets
D_{ALARM}	end-to-end delay experienced by delivered ALARM packets

Table 7.1. Protocol Performance Model Parameters.

models in Section 7.6.

7.1 Model of Reliability - LAR

We assume that when a route breaks, all packets currently at intermediate nodes are dropped in LAR. With this assumption, the data packet delivery ratio is

$$R = \frac{\alpha T_{route} - P_{lost}}{\alpha T_{route}}. \tag{7.1}$$

The number of packets sent on the route is αT_{route}, therefore the primary difficulty is determining how many packets are lost when a route breaks, P_{lost}. The number of packets lost depends on the route length, L, the packet arrival rate, α, and the per link delay, D_u.

D_u is dependent on the Medium Access Control and physical layer protocols; our simulations use IEEE 802.11. Thus, our simplified determination of D_u is

$$D_u = T_{tx}(1 + \sum_{i=1}^{7} {P_f}^i) \tag{7.2}$$

where T_{tx} is the time to transmit one data packet over one hop, and P_f is the probability of channel access failure. The unicast packet transmit time includes all link layer protocol overhead such as Request To Send (RTS), Clear To Send (CTS), and acknowledgment (ack). The additional time $\sum_{i=1}^{7} {P_f}^i$ represents the time spent in

back-off and transmission retry due to contention and congestion. In order to simplify our calculation, we set the probability of channel access failure, P_f, to $1 - R$. That is, we assume nodes fail to acquire the transmission channel, or experience collisions due to congestion, the same percentage of time that data packets are lost in route to the destination. We use the unicast delay, D_u, in subsequent modeling equations.

At a minimum, one packet is lost for every route error, regardless of which link on the route breaks. If a link on the route close to the destination breaks, then more packets will be both on the route at the time of the break and sent after the link breaks but before the route error is reported to the source. In the same way, if a link close to the source breaks, then fewer packets will be lost. In both cases, the values of L, α, and D_u contribute to the number of packets lost when a route breaks. We derive the number of packets lost when a route breaks to be

$$P_{lost} = \left\lceil \sum_{i=2}^{[L]} (2D_u(i-1)) + 1 \right\rceil .$$

7.2 Model of Reliability - ALARM

The primary difference between LAR and ALARM is that ALARM does not automatically drop data packets at intermediate nodes when a link breaks. ALARM attempts to "save" data packets by (box) flooding them toward the destination. As

a result, the expected data packet delivery ratio for ALARM becomes

$$R = \frac{\alpha T_{route} - (1 - F_r)P_{lost}}{\alpha T_{route}}, \tag{7.3}$$

where F_r is the delivery ratio for flooded packets. Unlike LAR which drops P_{lost} packets when a route error occurs, ALARM is able to deliver F_r of them successfully. In other words, ALARM drops only $(1 - F_r)P_{lost}$ data packets.

Figure 7.1 displays the simulated delivery ratios for LAR and ALARM and predicted delivery ratios for LAR and ALARM from Equations 7.1 and 7.3 respectively. LAR simulated and LAR predicted have a correlation coefficient of 0.936, and ALARM simulated and ALARM predicted have a correlation coefficient of 0.927.

We note that F_r in Equation 7.3 is calculated as

$$F_r = \frac{(N - O_{total})}{N} F_{raw},$$

where N is the number of nodes in the network, F_{raw} is the data packet delivery ratio for flooded data packets when no other network traffic is present, and O_{total} is the total overhead added by some other protocol operating in the network, in this case ALARM. In other words, we adjust the raw flooding reliability by the congestion added due to protocol operation. The total packet overhead per data packet delivered of flood trends toward N since every data packet is transmitted once by every node

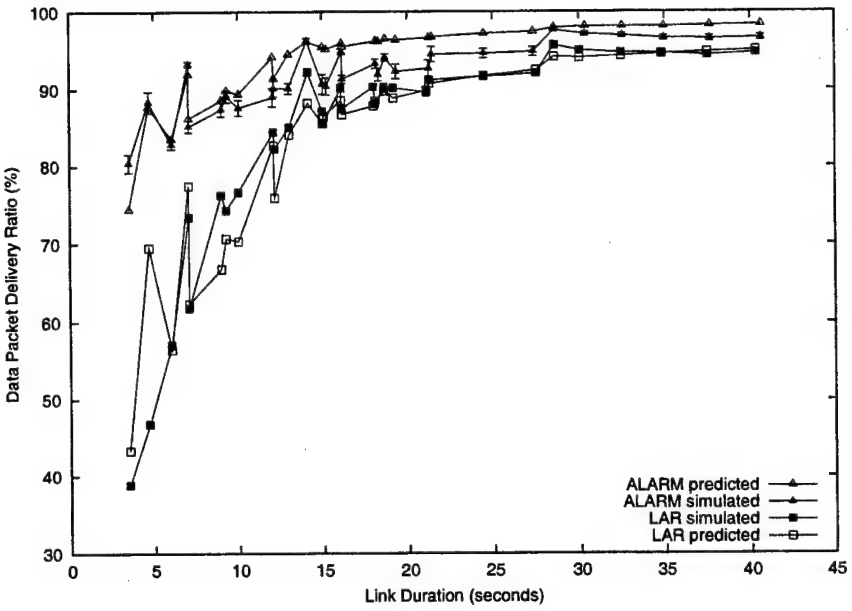

FIG. 7.1. Measured and Predicted Data Packet Delivery Ratio vs. Link Duration - LAR and ALARM
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

in the network. Flooding reliability is then reduced by the presence of other network traffic (protocol and data) by $\frac{N-O_{total}}{N}$. We use the flooding reliability, F_r, in the subsequent modeling equations.

7.3 Model of Protocol Overhead

Protocol overhead in LAR and ALARM is contributed by three main components

1. route requests,
2. route replies, and
3. route errors.

As discussed in Section 6.6, since the raw protocol overhead for LAR and ALARM is identical, one model of overhead is applicable to both. Differentiation in protocol overhead per data packet delivered is provided by the differing data packet delivery ratios. Route requests are typically flooded (or box flooded) to the destination and contribute

$$N(\frac{T_{sim}}{T_{route}})C + NC,$$

where T_{sim} is the simulation time, to overhead. In other words, a flooded route request packet is relayed by all N nodes in the network, every $\frac{T_{sim}}{T_{route}}$ seconds, for all C flows. The additional NC accounts for the initial route request for each flow. Route replies are unicast packets and contribute

$$L(\frac{T_{sim}}{T_{route}})C + LC$$

to overhead on average. In other words, route replies are unicast over L links, every $\frac{T_{sim}}{T_{route}}$ seconds, for all C flows and are unicast once for the reply to each initial route request. Lastly, route errors contribute

$$L(\frac{T_{sim}}{T_{route}})C$$

to overhead in the worst case.

To determine the number of overhead packets sent per data packet delivered, we use

$$RC\alpha T_{sim}$$

as the number of data packets successfully delivered. In other words, αT_{sim} packets for C flows with a success of R are sent. Combining overhead terms yields

$$\frac{N(\frac{T_{sim}}{T_{route}})C + NC + L(\frac{T_{sim}}{T_{route}})C + LC + L(\frac{T_{sim}}{T_{route}})C}{RC\alpha T_{sim}},$$

or more simply

$$\frac{2L+N}{T_{route}R\alpha} + \frac{L+N}{R\alpha T_{sim}}.$$

As the simulation time increases, the second term, which represents the initial route request/reply, trends to zero leaving

$$\frac{2L+N}{T_{route}R\alpha}. \tag{7.4}$$

One final source of overhead is the retransmission of a route request if the first attempt fails. Incorporating this overhead into Equation 7.4 yields

$$\frac{2L+(2-F_r)N}{T_{route}R\alpha}. \tag{7.5}$$

That is, route requests fail $(1-F_r)$ percent of the time. A second route request adds another N overhead transmissions this percent of the time. Using Equation 7.5, Figure 7.2 shows the simulated packet overhead of LAR and ALARM per data packet delivered and the expected overhead for LAR and ALARM per data packet delivered. We note that R in Equation 7.5 is R_{LAR} or R_{ALARM} depending on which protocol overhead is being predicted in Figure 7.2. LAR (ALARM) simulated and LAR (ALARM) predicted have a correlation coefficient of 0.990 (0.942) for packet overhead per data packet delivered.

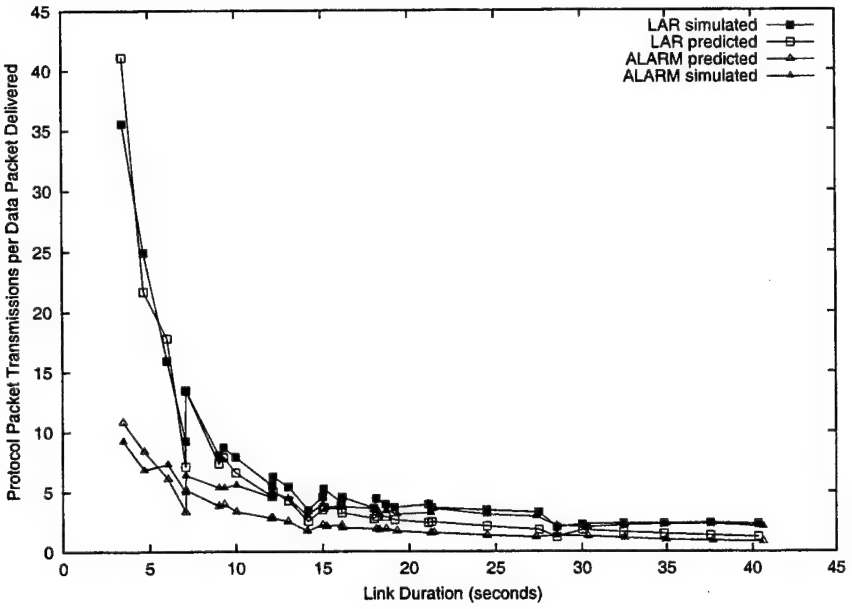

FIG. 7.2. Measured and Predicted Protocol Packet Transmissions per Data Packet Delivered vs. Link Duration - LAR and ALARM
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

7.4 Model of Delay - LAR

Delay occurs in LAR as a result of the protocol operation and the transit time of the data packet. We consider the transit time of the data packet to be $D_u L$, where D_u is the single hop unicast packet delay which is multiplied by the average route length. Protocol operation adds delay for route requests, replies, and errors. For

flooded route requests we use a delay of $D_f L$, where D_f is the single hop flooded packet delay. For unicast route replies and errors we use the unicast delay $D_u L$. The determination of D_f is taken from prior experiments with flooding. In our case this value is 0.11 seconds on average. D_u is determined using Equation 7.2.

We define a base end-to-end delay, without queue waiting time, for each flow per data packet delivered as

$$D_{base} = D_u L + \frac{D_f L + 2D_u L + (1 - F_r)(D_f L + D_u L + 0.5)}{T_{route} R \alpha}. \tag{7.6}$$

Equation 7.6 represents the data packet transit time ($D_u L$), plus the route request, route reply, and route error time ($D_f L + 2D_u L$) averaged over all the data packets that will traverse that route before it breaks ($T_{route} R\alpha$). The added term accounts for the failure of the initial route request ($1 - F_r$) percent of the time. Thus, another route request/reply cycle is needed plus the route request time out delay. In LAR (and ALARM), the route request time out period is $\frac{1}{2}$ second.

One final source of delay, which primarily occurs in high mobility situations, is due to packets waiting in the pending packet queue. When mobility is high the route request/reply cycle (box flood, timeout, full flood) can fail entirely. When this happens the pending packets are placed in a pending packet queue. Whenever a route request is successful for a destination, the pending packet queue is searched for packets awaiting transmission for the destination. These packets are then sent to the

destination.

Our implementation of LAR uses a 64 packet queue which deletes packets older than 30 seconds. We note that our data traffic model generates four packets per second, α. Therefore pending packets are dropped from the queue in our simulations after only 16 seconds. These implementation specifics lead to a queue waiting time of

$$Q_{wait} = \frac{Q_{average}(1 - R)}{T_{route}} \tag{7.7}$$

where $Q_{average}$ is the average queue waiting time, which in our case is eight seconds. The queue waiting time is influenced heavily by short link durations and low data packet delivery. Thus we adjust the queue waiting time by $\frac{1-R}{T_{route}}$. Our final expression for end-to-end delay in LAR is

$$D_{LAR} = D_{base} + Q_{wait}. \tag{7.8}$$

Equation 7.8 simply combines Equations 7.6 and 7.7 to calculate total delay.

7.5 Model of Delay - ALARM

Our model for ALARM delay builds upon the LAR delay model shown in Equation 7.8. When mobility is low, most packets travel from source to destination using only the mechanisms of LAR. As mobility moves higher, ALARM mechanisms are

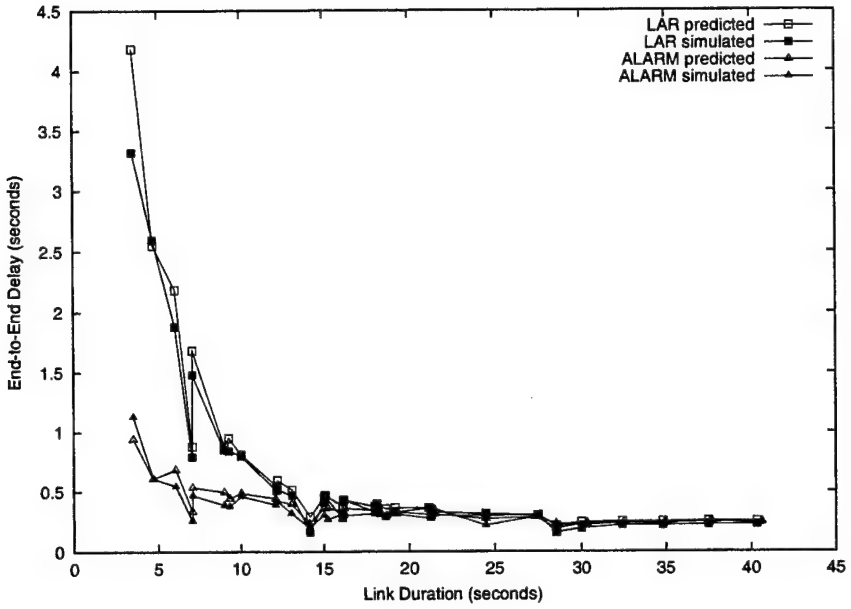

FIG. 7.3. Measured and Predicted End-to-End Delay vs. Link Duration - LAR and ALARM
Link duration = 16.125 seconds, speed = 10 m/s, pause time = 10 seconds.
100m transmission range, 300x600m area, 50 mobile nodes.
20 communicating pairs, 4 packets per second, 64 bytes per packet.
Beacon period = 1 second, link break time = 3 seconds,
duration window size = 30 seconds.
ALARM threshold = 9 seconds, flood horizon = 1 hop.

used to deliver more data packets. Thus, our model for delay in ALARM is

$$D_{ALARM} = \frac{R_{LAR}D_{LAR} + (R_{ALARM} - R_{LAR})D_f}{R_{ALARM}}. \tag{7.9}$$

Equation 7.9 combines the LAR delay, D_{LAR}, experienced by $\frac{R_{LAR}}{R_{ALARM}}$ delivered data packets with the flooded packet delay, D_f, experienced by the $\frac{R_{ALARM}-R_{LAR}}{R_{ALARM}}$ "saved"

data packets. Figure 7.3 shows the simulated end-to-end delay and the expected end-to-end delay for both LAR (Equation 7.8 from Section 7.4) and ALARM (Equation 7.9). LAR (ALARM) simulated and LAR (ALARM) predicted have a correlation coefficient of 0.991 (0.948) for end-to-end delay.

7.6 Conclusions

The models developed in this chapter aid us in validating our simulation results. In addition, we obtain an added understanding that comes from the analytical study of protocol operation. For example, our models emphasize the dependence which both LAR and ALARM, and reactive protocols in general, have on T_{route}. ALARM additionally shows a dependence on flooding effectiveness, F_r, as it depends on flooding to deliver data packets lost by LAR due to route errors. In other words, ALARM's reliability is heavily dependent on F_r. Figure 7.1 illustrates that our adjustment to F_{raw} is not completely accurate; i.e., our ALARM prediction of reliability often over-predicts the reliability provided by our ALARM simulation.

Prediction of protocol overhead is dependent on and sensitive to F_r as well. This proves to be the dominant element of Equation 7.5 since the flooding of route requests in reactive protocols is their most expensive component. We note, however, that our flooding model for ALARM predicts the overhead effectively (see Figure 7.2).

Finally is our model and estimation of end-to-end delay. This model is heavily

dependent on the link and physical layer protocols being used. The input parameters D_f and D_u are both determined primarily by the operation of the lower layers. Our model shows good prediction when the network is stable (see Figure 7.3) as delay for LAR and ALARM converge. The addition of queue waiting time (see Equation 7.7) allows us to also model delay well in times of high mobility.

Chapter 8

CONCLUSIONS

MANET routing protocols have been extensively studied (see Section 1.3); however, two key goals remain elusive. One, no existing routing protocol has demonstrated effective operation in a wide range of network dynamics (contention, congestion, and mobility). Two, applying existing routing protocols to larger networks shows extensive scalability difficulties. Our research targets both challenges through the use of location information and mobility feedback, which allows us to create a hybrid, adaptive MANET routing protocol. Our hybrid, adaptive protocol performs significantly better than previously proposed MANET routing protocols.

In Chapter 2 we examine the use of location information in MANET routing. We demonstrate that location information is beneficial to protocol operation as it increases the percentage of data packets delivered and lowers delay. We also illustrate that there is a nominal cost of increased overhead in order to achieve this improvement in both reliability and end-to-end delay. Specifically, we include five major conclusions (see Section 2.5).

1. The added protocol complexity of DREAM [2] does not appear to provide benefits over network wide flooding of data packets.

2. Location information improves DSR [34], especially at high speeds.

3. Promiscuous mode operation improves the performance of DSR significantly.

4. Our implementation of DREAM provides a simple location service.

5. There is a tradeoff between average end-to-end delay and data packet delivery ratio.

To obtain the benefit of location information we require a mechanism to distribute node locations within a MANET. In Chapter 4, we propose and evaluate three methods to deliver location information. Our proposed methods have parallels with existing wired, wireless, and MANET routing protocols. We propose and evaluate (via simulation) the following location services for an ad hoc network.

1. Reactive Location Service (RLS): node locations are requested and discovered on demand similar to DSR [34].

2. Simple Location Service (SLS): node location tables are exchanged between neighboring nodes similar to RIP [26].

3. DREAM's Location Service (DLS): individual node locations are broadcast to neighbors frequently and to the entire network with lower frequency similar to OSPF [40].

Our evaluation of the proposed protocols shows a performance advantage of SLS over DLS and RLS.

Our drive to enable protocol adaptivity required a suitable mobility metric. In Chapter 3 we present requirements for a mobility metric and demonstrate that link duration meets these requirements (see Section 3.3). Once identified as an accurate mobility metric, it became necessary to provide link duration information to protocols operating on mobile nodes. Therefore, in Chapter 5, we discuss the design, implementation, and evaluation of a feedback agent which provides link duration information in a distributed manner. Our feedback agent has no added overhead when data traffic is present. When data traffic is not present, providing effective link duration information is not necessary to increase performance of the overall system. However, our feedback agent can actively collect link duration information if desired.

Our research culminates in the design and evaluation of a MANET routing protocol which exploits the use of location information and mobility feedback by fusing two existing protocols into an adaptive hybrid protocol (see Chapter 6). ALARM out performs both component protocols which is shown by the increased data packet delivery ratio and decreased protocol overhead for all mobility scenarios tested. The increased reliability and decreased overhead is especially apparent in demanding, high mobility network scenarios. In addition to strong performance, ALARM demonstrates the effectiveness of both adaptive protocol operation and the adaptive combination of existing protocols.

One final contribution is the development of analytical models for reliability,

overhead, and delay of both LAR and ALARM. Our models support link duration as an effective mobility metric, validate the simulated results presented, and facilitate a deeper understanding of protocol operation. The predictive accuracy of the models is quite high. None of the six models has a correlation coefficient under 0.92.

Mobile ad hoc networks embody many challenges, and promise great freedom for every user of mobile computing devices. This dissertation helps move us another step closer to the vision of ubiquitous computing.

REFERENCES

[1] S. Basagni. Personal communication. November 2000.

[2] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the Fourth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 1998)*, pages 76–84, 1998.

[3] B. Bellur, R. Ogier, F. Templin, and M. Lewis. Topology broadcast based on reverse-path forwarding (TBRPF). Internet Draft: draft-ietf-manet-tbrpf-05.txt, March 2002.

[4] J. Boleng. Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks. In *Proceedings of LANMAN 2001: 11th IEEE Workshop on Local and Metropolitan Area Networks*, pages 9–12, March 2001.

[5] J. Boleng, T. Camp, and W. Navidi. Metrics to enable adaptive protocols for mobile ad hoc networks. In *Proceedings of the International Conference on Wireless Networking (ICWN)*, pages 293–298, 2002.

[6] J. Boleng, T. Camp, and V. Tolety. Mesh-based geocast routing protocols in an ad hoc network. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS): Workshop on Parallel and Distributed*

Computing Issues in Wireless Networks and Mobile Computing (PDC), April 2001.

[7] J. Boleng, B. Williams, T. Camp, L. Wilcox, and W. Navidi. Performance of location-based routing protocols for an ad hoc network. In *Proceedings of LANMAN 2001: 11th IEEE Workshop on Local and Metropolitan Area Networks*, pages 98–101, March 2001.

[8] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 1998)*, pages 85–97, 1998.

[9] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC)*, Special Issue, 2002. To appear.

[10] T. Camp, J. Boleng, and L. Wilcox. Location information services in mobile ad hoc networks. In *Proceedings of the IEEE International Communications Conference (ICC)*, pages 3318–3324, 2002.

[11] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance comparison of two location based routing protocols for ad hoc networks. In

Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002), pages 1678–1687, 2002.

[12] C. Chiang, H.K. Wu, W. Liu, and M. Gerla. Routing in clusterhead multihop, mobile wireless networks with fading channel. In *Proceedings of the IEEE Singapore International Conference on Networks (SICON '97)*, pages 197–211, 1997.

[13] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (OLSR). Internet Draft: draft-ietf-manet-olsr-06.txt, September 2001.

[14] S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM Journal on Wireless Networks*, 1(1):61–81, 1995.

[15] R. Dube, C.D. Rais, K.-Y. Wang, and S.K. Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications*, pages 36–45, February 1997.

[16] The IEEE Working Group for Wireless Local Area Network Standards. IEEE wireless local area networks. http://grouper.ieee.org/groups/802/11/. Page accessed on August 14, 2002.

[17] M. Gerla, X. Hong, L. Ma, and G. Pei. Landmark routing protocol (LANMAR)

for large scale ad hoc networks. Internet Draft: draft-ietf-manet-lanmar-04.txt, June 2002.

[18] M. Gerla, X. Hong, and G. Pei. Fisheye state routing protocol (FSR) for ad hoc networks. Internet Draft: draft-ietf-manet-fsr-03.txt, June 2002.

[19] M. Gerla, G. Pei, and S.-J. Lee. Wireless, mobile ad-hoc network routing. In *Proceedings of the IEEE/ACM Federation on Computing in the United States (FOCUS)*, 1999.

[20] T. Goff, N. Abu-Ghazaleh, D. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *Proceedings of the Seventh Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2001)*, pages 43–52, 2001.

[21] Marc Greis and The VINT Group. Tutorial for the network simulator - ns. http://www.isi.edu/nsnam/ns/tutorial/index.html. Page accessed August 14, 2002, 2000.

[22] Z. Haas. A new routing protocol for reconfigurable wireless networks. In *Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC)*, pages 562–565, Oct. 1997.

[23] Z. Haas, M. Pearlman, and P. Samar. The interzone routing protocol (IERP) for ad hoc networks. Internet Draft: draft-ietf-manet-zone-ierp-01.txt, June 2001.

[24] Z. Haas, M. Pearlman, and P. Samar. The intrazone routing protocol (IARP) for ad hoc networks. Internet Draft: draft-ietf-manet-zone-iarp-01.txt, June 2001.

[25] Z. Haas, M. Pearlman, and P. Samar. The bordercast resolution protocol (BRP) for ad hoc networks. Internet Draft: draft-ietf-manet-zone-brp-02.txt, July 2002.

[26] C. Hedrick. Routing information protocol. Request for Comments 1058, June 1988.

[27] M. Horner and D. Plassmann. Directed antennas in the mobile broadband system. In *Proceedings of the 15th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 1996)*, pages 704–712, 1996.

[28] Y.-C. Hu and D. B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 231–242, 2000.

[29] P. Jacquet and A. Laouiti. Analysis of mobile ad-hoc network routing protocols in random graph models. Rapport de recherche 3835, INRIA, 1999.

[30] P. Jacquet and L. Viennot. Overhead in mobile ad-hoc network protocols. Rapport de recherche 3965, INRIA, 2000.

[31] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial informa-

tion for wireless ad hoc networks. *IEEE Personal Communications*, pages 48–57, February 2001.

[32] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Routing protocols for mobile ad-hoc networks - a comparative performance analysis. In *Proceedings of the Fifth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 1999)*, pages 195–206, 1999.

[33] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imelinsky and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[34] D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft: draft-ietf-manet-dsr-07.txt, February 2002.

[35] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 243–254, 2000.

[36] Y. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the Fourth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 1998)*, pages 66–75, 1998.

[37] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 120–130, 2000.

[38] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu. Grid: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1):37–60, 2001.

[39] J. Macker and S. Corson (Chairs). Mobile ad hoc networks (MANET). http://www.ietf.org/html.charters/manet-charter.html. Page accessed August 14, 2002., 1997.

[40] J. Moy. OSPF version 2. Request for Comments 2178, July 1997.

[41] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[42] S. Murthy and J.J. Garcia-Luna-Aceves. A routing protocol for packet radio networks. In *Proceedings of the First Annual ACM International Conference on Mobile Computing and Networking (MobiCom 1995)*, pages 86–95, 1995.

[43] V. Park and S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the 16th Annual Joint Conference*

of the IEEE Computer and Communications Societies (Infocom 1997), pages 1405–1413, April 1997.

[44] V. Park and S. Corson. Temporally-ordered routing algorithm (TORA) version 1 functional specification. Internet Draft: draft-ietf-manet-tora-spec-04.txt, July 2001.

[45] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on demand distance vector (AODV) routing. Internet Draft: draft-ietf-manet-aodv-11.txt, June 2002.

[46] C. Perkins and P. Bhagwat. Destination sequenced distance vector routing for mobile computers. *Computer Communication Review: SIGCOMM*, 24(4):234–244, October 1994.

[47] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, pages 234–244, 1994.

[48] C. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100, 1999.

[49] Charles E. Perkins, editor. *Ad Hoc Networking.* Addison-Wesley, 2001.

[50] The Rice Monarch Project. The Rice monarch extensions to the ns simulator. http://www.monarch.cs.rice.edu/cmu-ns.html. Page accessed on August 14, 2002.

[51] E. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[52] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.

[53] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications*, 4(2):1–36, 1997.

[54] C.-K. Toh. Long-lived ad hoc routing based on the concept of associativity. Internet Draft: draft-ietf-manet-longlived-adhoc-routing-00.txt, March 1999.

[55] Y.-C. Tseng, S.-L. Wu, W.-H Liao, and C.-M. Chao. Location awareness in ad hoc wireless mobile networks. *Computer*, 34(6):46–52, 2001.

[56] B. Williams, J. Boleng, and T. Camp. Geocast communications in an ad hoc network. In *Proceedings of LANMAN 2001: 11th IEEE Workshop on Local and Metropolitan Area Networks*, pages 94–97, March 2001.

[57] T.-S. Yum and K.-W. Hung. Design algorithms for multihop packet radio networks with multiple directional antennas stations. *IEEE Transactions on Communications*, 40(11):1716–1724, 1992.